Prediction of Protein Structures Based on Curve Alignment *

Yi-Ying Chen, Chang-Biau Yang and Kuo-Tsung Tseng
Department of Computer Science and Engineering
National Sun Yat-sen University, Kaohsiung, Taiwan
cbyang@cse.nsysu.edu.tw

Abstract

The biochemical functions of proteins are determined by their structures. Thus one of the most important issues in the life science is to predict the three-dimensional structures with protein sequences, and then to deduce their biochemical functions. In order to simplify the problems, scientists use the lattice model to approximate the real protein structures, but they two cannot be compared in fact. So we present the curve fitting concept, such as B-splines, to convert the lattice model and a real structure to the curves to see the difference among them in a fair position. Besides, the curve alignment can also be used as another measurement to evaluate the similarity between two real protein structures. We then propose an algorithm to develop a protein structure prediction methodology based on a structure-known protein, where the two protein sequences are extremely similar. By the experimental results, our protein structure prediction method performs well when we get two protein sequences with similarity that is not too high.

Key words: computational biology, protein structure, prediction, lattice, spline

1 Introduction

Proteins are macromolecules that perform all important tasks in organisms, such as catalysis of chemical reactions. It is widely accepted that three-dimensional structure of proteins determine their functions. Traditionally, the tertiary structures have been solved by X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy [22, 28, 16]. However, they are often difficult to be crystallized and time-consuming. Therefore, protein three-dimensional structure prediction from the amino acid sequence is very important in bioinformatics.

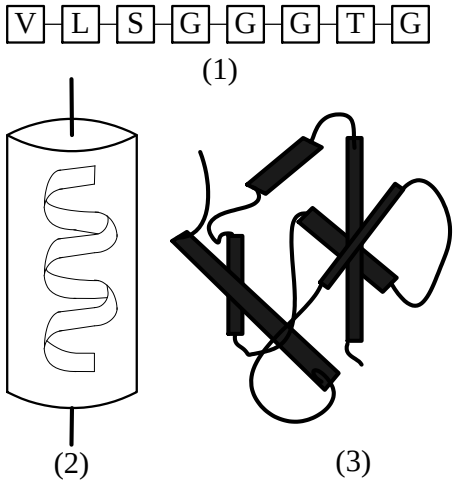


Figure 1: Protein primary, secondary and tertiary structures.

Proteins are chains of amino acid residues. A protein sequence consists of twenty different kinds of amino acids, so called the *primary structure*. The *secondary structure* of a protein corresponds to regions of local regularity such as α-helix and β-sheet. The *tertiary structure* of a protein arises from the packing of its secondary structure elements, which may be from discrete domains within a fold, or may give rise to autonomous folding units or modules, complete folds, domains and modules. Every protein has a unique primary sequence specifying its tertiary structure. Figure 1 illustrates these structures.

It is believed that two protein sequences diverged from the same ancestor show a certain degree of similarity. The sequence *similarity* is usually used to measure the genetic distance between two proteins. Given a sequence, another sequence can be generated by randomly inserting, deleting or replacing some characters on the first sequence. The degree of similarity of two sequences means the number of the above operations

*This research work was partially supported by the National Science Council of the Republic of China under contract NSC-90-2213-E-110-022.

applied on the first sequence to become the other one. Therefore, we obtain the similarity by aligning two sequences. Dynamic programming algorithms are often used to generate the optimal pairwise alignment[2]. If an alignment has more than 25% $\sim$ 30% sequence similarity, it is generally assumed that two sequences have diverged from the same ancestor[17, 4]. However, the two proteins may still diverge from the same ancestor but they are highly different.

To determine the conformation of a given protein sequence is called the *protein folding problem*, which is one of the protein structure prediction problems. This problem has been studied since 1950s[5, 19, 20], but there is no satisfactory solution yet. Nowadays, common approaches used to predict the structure of proteins are *homology modelling, fold recognition*, and *ab initio*[23].

Several researches have studied the protein folding problem in the *lattice model* and developed various kinds of folding algorithms[12, 24] to predict real protein conformations. However, no studies have been done in comparing the folding conformation and the real structure. We now propose a structure alignment method with curve fitting to compare the conformation generated by the folding algorithm in the lattice with the real protein structure and an algorithm to solve the protein prediction problem.

The rest of this paper is organized as follows. In Section 2, we first introduce the *Hydrophobic-hydrophilic*(HP) lattice model and some folding algorithms are introduced. Next, we introduce the B-Spline curve fitting. In Section 3, we present our method. In Section 4 and 5, the experimental results and conclusions are given, respectively.

2 Preliminaries

2.1 The Hydrophobic-hydrophilic Model

The *hydrophobic-hydrophilic model* proposed by Dill is a lattice model[7]. In this model, the protein sequence is abstracted by *hydrophobic* (non-polar, "water-hating" or "water-disliking") and *hydrophilic* (polar, "water-loving" or "water-liking") residues. Table 2.1 shows the hydrophobic or hydrophilic property for each of twenty amino acids. Thus it can be read as a string over the alphabets $\{0,1\}$ where 1 represents "H" and 0 represents "P". This model is often to be referred as the *HP model*, where H represents "hydrophobic" and P represents "polar". From experimental data, hydrophobic residues tend to form the protein core and the polar ones tend to cover over the surface during the folding process.

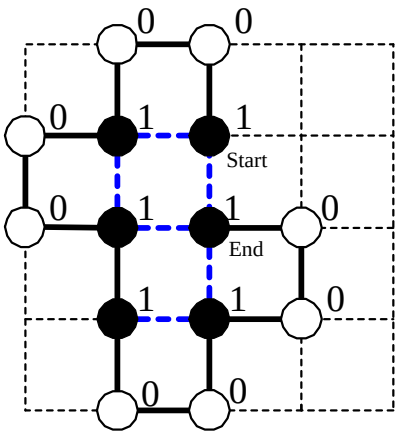


Figure 2: A conformation in 2D HP model with energy 6.

A *conformation* in a HP lattice model is an embedding of the protein sequence (chain of *beads*) in some lattice, and it is modelled as a *self-avoiding* path. In a self-avoiding path, every bead in the chain occupies a lattice site, but no two beads occupy the same lattice site. Two beads that are neighbors in the chain occupy adjacent lattice sites.

The free energy of a conformation depends on the number of non-neighboring hydrophobic amino acids that occupy adjacent grid points in the lattice. Each hydrophobic-hydrophobic (H-H) bond contributes one free energy of E ($E < 0$), and hydrophobic-hydrophilic (H-P) and hydrophilic-hydrophilic (P-P) bonds would have free energy of 0. Theoretically, the *native conformation* is the one with minimal free energy. That is, it maximizes the number of contacts between hydrophobic residues. Figure 2 shows a conformation in the 2D HP model where 6 non-neighboring hydrophobic amino acids occupy adjacent grid points. Let S denote the abstraction of an amino acid sequence of length n. $S[i]$ is 1 if the ith amino acid in the sequence is hydrophobic and 0 if it is hydrophilic. Figure 2 illustrates a protein sequence 10010011001001 in the 2D HP model.

Though the HP model is quite simple, it is powerful enough to capture the properties of proteins. The protein structure prediction problem for the HP model has been proved to be NP-complete in the square lattice [3, 6, 9, 14, 26]. Many approximation algorithms have been developed for the HP lattice model[1, 12, 13, 27, 18].

2.2 B-Spline Curve Fitting

The *B-Spline curve* had been devoted by Lobachevsky in the nineteenth century and is one

Table 1: Twenty naturally occurring amino acids found in biological systems.

	One-letter code	Three-latter code	Name	**H**ydrophobic or hydrophilic(**P**)
1	A	Ala	Alanine	H
2	C	Cys	Cysteine	H
3	D	Asp	Aspartic Acid	P
4	E	Glu	Glutamic Acid	P
5	F	Phe	Phenylalanine	H
6	G	Gly	Glycine	H or P
7	H	His	Histidine	P
8	I	IlE	Isoleucine	H
9	K	Lys	Lysine	P
10	L	Leu	Leucine	H
11	M	Met	Methionine	H
12	N	Asn	Asparagine	P
13	P	Pro	Proline	H
14	Q	Gln	Glutamine	P
15	R	Arg	Arginine	P
16	S	Ser	Serine	P
17	T	Thr	Threonine	P
18	V	Val	Valine	H
19	W	Trp	Tryptophan	H
20	Y	Tyr	Tyrosine	H

type of spline, perhaps the most popular, in computer graphics applications [8, 10]. A B-Spline curve is a set of piecewise (usually cubic) polynomial segments that pass close to a set of control points. The curve is formed in relation to the 3D polyline joining the points in sequence. The B-Spline Curve always starts at the first control point and ends at the last control point, and it is always tangent to the polyline at these end points, but in general it does not pass through the other control points.

The B-Spline curve is defined as follows: for given $n+1$ control points $P_0, P_1, P_2, \ldots, P_n$, we can derive a continuous function $P(v)$ as

$$P(v) = \sum_{k=0}^{n} P_k N_{k,t}(v)$$

where $N_{k,t}(v)$ is a *blending function*, and t is the degree of the polynomials for representing a curve segment that is usually 3 or 4.

There are a number of possible options for the knot positions, for example, a uniform spacing where $u[k] = k$. More commonly the following function is chosen

$$u[k] = \begin{cases} 0 & 1 \le k < t \\ k-t+1 & t \le k \le n \\ n-t+2 & k > n \end{cases}$$

The blending functions determine how strongly control point P_k influences the curve at point v, which are defined as

$$N_{k,\,1}(v) = \begin{cases} 1 & if\ u[k] \le v < u[k+1] \\ 0 & otherwise \end{cases}$$

$$\begin{aligned} N_{k,\,t}(v) &= \frac{v-u[k]}{u[k+t-1]-u[k]} N_{k,\,t-1}(v) \\ &+ \frac{u[k+t]-v}{u[k+t]-u[k+1]} N_{k+1,\,t-1}(v) \end{aligned}$$

Next we present a previous approach to find a correspondence (alignment) between two curves[11, 21]. The correspondence is based on a notion of an alignment which treats both curves symmetrically.

Let $C_1(s_1) = (x_1(s_1), y_1(s_1))$, $s_1 \in [0, L_1]$, and $C_2(s_2) = (x_2(s_2), y_2(s_2))$, $s_2 \in [0, L_2]$, denote two curve segments, where s denotes one point, x and y are the coordinates of each point, and L represents the length of the projection of the curve segment on the x-axis. Consider two curve segments $C_{1[A_1,B_1]}$ and $C_{2[A_2,B_2]}$ of lengths ds_1 and ds_2, respectively. A mapping $g : [0, L_1] \to [0, L_2]$, $g(s_1) = s_2$ represents an alignment of the two curve segments. Therefore, we can align the two curves such that two points A_1 and A_2, the tangents T_1 and T_2 are overlapping, as shown in Figure 3. The cost of the alignment is the combination

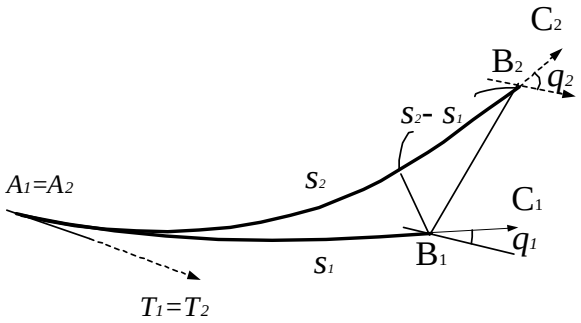


Figure 3: The cost of two curve segments $C_1 = A_1B_1$ and $C_2 = A_2B_2$.

of degree of points B_1 and B_2 and their tangents. Thus, a measure μ on this function[21] is defined as

$$\mu = |ds_2 - ds_1| + r|dq_2 - dq_1|,$$

where r is a constant.

3 A Prediction Method Based on Curve Alignment

In this section, we shall propose a heuristic algorithm to solve the protein prediction problem. Given two similar protein sequences and the structure of one of the two sequences, we are desired to predict the structure of the other one. Our algorithm is the combination of the homology modelling method and the folding algorithm.

It is clear that the HP lattice model and the real protein structure cannot be compared directly. In other words, they two must be transformed into both the HP lattice model or both the real protein structure to be used to predict or see the difference between them. It is difficult to have a transform like this. Thus we present the curve fitting concept, as a medium of two types. By making the HP lattice model and the real protein stucture into curves, they are easy to compared with and offer us some other candidates when we are trying to predict the tertiary structure of a protein.

In our algorithm, we apply the B-spline curve fitting in protein structure alignment because of the difficultly in comparing the conformation generated by the folding algorithm in the lattice with the real protein structure. Using B-spline curves, we can use the curve alignment method and easily determine which conformation is closer. There are many kinds of spline curves. The reasons we use B-spline curves are the properties described in Section 2.2 and these properties are closer to real protein structures than those of other spline curves.

Definition 1 *Given two protein sequences (the master sequence and slave sequence) and the tertiary structure information of the master sequence, the protein structure prediction problem is to predict the tertiary structure of the slave sequence.*

Definition 2 *Given two protein sequences with higher than 30% sequence similarity, we can say that they are homologous. Thus we may say that they evolve from the same ancestor and hence are highly structurally related[4].*

Definition 3 *The structurally conserved regions are those sequences of residues in a structure-unknown protein which are highly homologous with those in a known structure.*

Our prediction algorithm is as follows.

Algorithm: Homology Modelling in Folding Algorithm

Input: Two protein sequences S_1 and S_2, where the structure of S_2 is known and S_1 is highly similar to S_2 with respect to their sequences.

Output: The backbone conformation model of S_1.

Step 1: Perform sequence alignment on S_1 and S_2.

Step 2: Find the structurally conserved regions, which have 50% or higher sequence similarity and the sequence alignment score is positive. Copy the coordinators of structurally conversed regions, except gaps, in the template structure S_2 to the target protein structure S_1.

Step 3: On the lattice model, apply the folding algorithm to position the residues that lose sequence similarity.

Step 4: For each G-Region G_i, find the structure-known proteins with 70% or higher sequence similarity to G_i. Then, construct a segment of B-spline curve for every four points of the folding structure and the similar protein structures. Apply the curve alignment between the folding structure and the similar protein structures. Copy the coordinators from the similar protein structure that gets the highest score.

Step 5: Construct the complete protein structure backbone model of S_1.

We explain the above algorithm with the following example step by step. Let us consider the two homologous protein sequences S_1 and S_2, that is, S_1 and S_2 are highly structurally related:

S_1 = SSKCSRLKTFPQNACVYHK
S_2 = SVYCSSLACSDHN

Suppose S_1 is the protein whose structure we want to predict. At step 1, we align S_1 and S_2 with the score Matrix PAM-250. In the following, we use | to represent that residues are identical, ⋮ to represent that they are similar, and - to represent a gap.

S_1 = SSKCSRLKTFPQNACVYHK
|--||⋮|------||--|⋮
S_2 = SVYCSSL------ACSDHN

In step 2, find the structurally conserved regions:

S S K C S R L
| - - | | ⋮ |
S V Y C S S L

↑
Structurally Conserved Region

K T F P Q N
- - - - - -
- - - - - -

↑
G – Region

A C V Y H K
| | - - | ⋮
A C S D H N

↑
Structurally Conserved Region

Then, we copy the coordinates from the similar segments, as shown in Figure 4. Next, translate the residues that lose similarity to a 0/1 string, where 1 represents ”hydrophobic” and 0 represents ”hydrophilic” as follows.

LKTFPQNA $\Longrightarrow$ 10011001

Apply the folding algorithm to position these residues. The folding conformation is shown in Figure 5.

Now, we obtain the coordinates of the folding residues from the folding conformation. Apply the B-spline curve formula to depict the global similar curve, as shown in Figure 6.

Then, we apply the matching method introduced in Section 2.2. Let μ be the score of curve segment matching and d a predefined threshold. Here our score

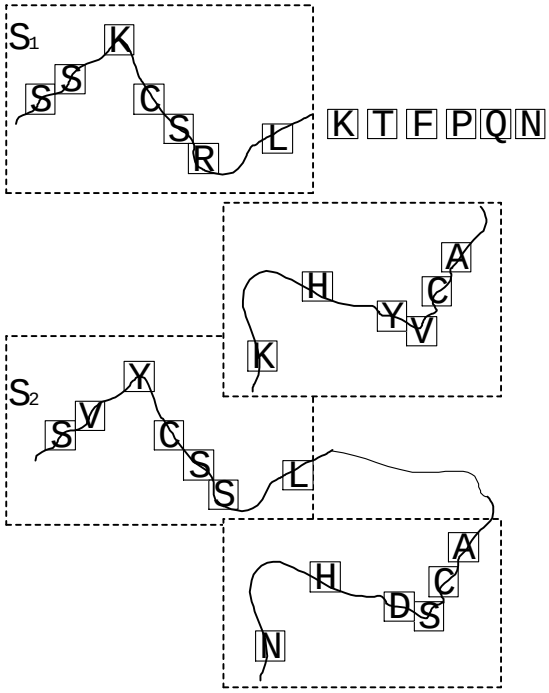


Figure 4: The process of cloning coordinates.

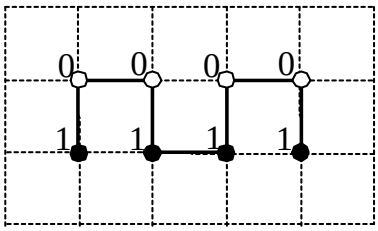


Figure 5: The folding conformation.

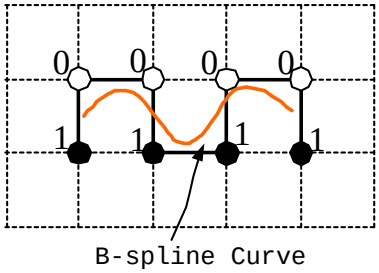
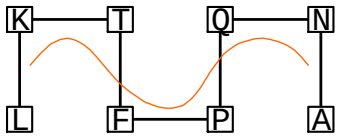


Figure 6: The B-spline curve of the folding conformation.

the folding structures



candidate protein structures

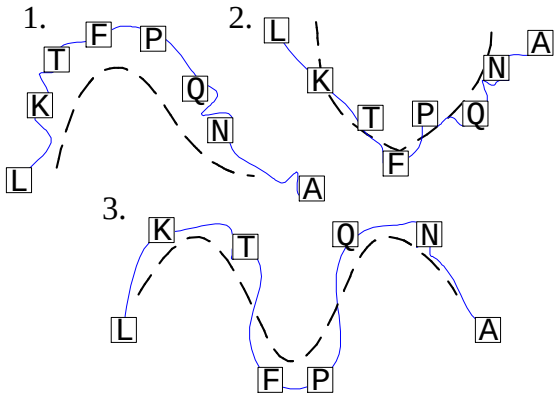


Figure 7: The three given candidate proteins, where broken lines represent the B-spline curve and thin lines represent the real conformation.

function of the curve alignment is defined as follows:

$$w_{i,j} = \begin{cases} +2 & if\ 0 < \mu < d\ (the\ matching\ is\ quite\ good) \\ +1 & if\ d \le \mu < 3d\ (the\ matching\ is\ good) \\ -1 & if\ 3d \le \mu < 6d\ (the\ matching\ is\ bad) \\ -2 & else\ \mu \ge 6d\ (the\ matching\ is\ too\ bad \\ & and\ a\ gap\ is\ inserted) \end{cases}$$

To compute the optimal structure alignment, we use the following dynamic programming, which is also used in the standard sequence alignment algorithm[25, 15].

$$A[i,j] = max(A[i-1,j], A[i,j-1], A[i-1,j-1] + w_{i,j})$$

By dynamic programming strategy, we can align two curves. Suppose that the given candidate proteins are shown in Figure 7. Then we get the B-spline curve of every four points from each of the candidate proteins and calculate the structure alignment matching score. Tables 2, 3 and 4 show the alignment score matrices between the folding conformation and one of the candidate proteins, respectively. Let X_i represent the B-spline curve composed of four residues starting at symbol X in the ith candidate protein and X_0 represent the curve starting at X in the folding structure. For example, T_2 represents the curve segment $TFPQ$ in candidate protein 2 and L_0 represents the curve $LKTF$ in the folding conformation.

Table 2: The structure alignment score matrix between candidate protein 1 and the folding structure.

	-	L_0	K_0	T_0	F_0	P_0
-	0	-2	-4	-6	-8	-10
L_1	-2	-1	0	-2	-4	-6
K_1	-4	-1	-2	1	-1	-3
T_1	-6	-3	-2	-1	0	-2
F_1	-8	-5	-1	-3	0	-1
P_1	-10	-7	-3	-2	-2	-1

Table 3: The structure alignment score matrix between candidate protein 2 and the folding structure.

	-	L_0	K_0	T_0	F_0	P_0
-	0	-2	-4	-6	-8	-10
L_2	-2	-1	-3	-5	-7	-9
K_2	-4	-3	-2	-4	-6	-8
T_2	-6	-5	-4	-3	-5	-7
F_2	-8	-7	-6	-5	-4	-6
P_2	-10	-9	-8	-7	-6	-5

Table 4: The structure alignment score matrix between candidate protein 3 and the folding structure.

	-	L_0	K_0	T_0	F_0	P_0
-	0	-2	-4	-6	-8	-10
L_3	-2	2	0	-2	-4	-6
K_3	-4	0	4	2	0	-2
T_3	-6	-2	2	6	4	2
F_3	-8	-4	0	4	8	6
P_3	-10	-6	-2	2	6	10

Table 5: The optimal structure alignment, where C_i represents the conformation of the ith candidate protein (i=0 represents the folding structure).

	Alignment	Cost
(C_0,C_1)	L_0 K_0 T_0 - F_0 P_0 - | ⋮ - ⋮ - - L_1 K_1 T_1 F_1 P_1	-1
(C_0,C_2)	L_0 K_0 T_0 F_0 P_0 - - - - - L_2 K_2 T_2 F_2 P_2	-5
(C_0,C_3)	$L_0K_0T_0F_0P_0$ | | | | | $L_3K_3T_3F_3P_3$	10

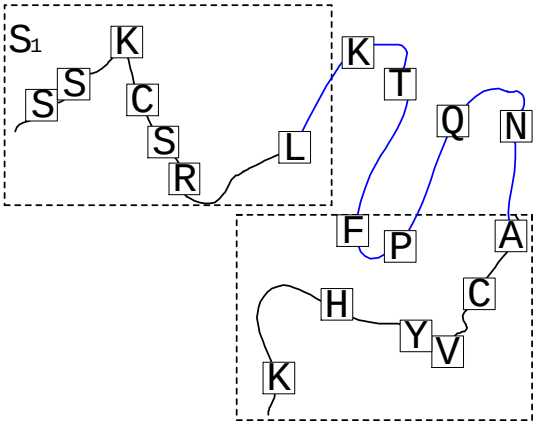


Figure 8: The final conformation of the structure-unknown protein S_1.

We can find the candidate protein with the highest score in the structure alignment. Table 5 shows the optimal structure alignments with the three candidates. Note that we get two pairs of similar regions (K_0T_0, L_1K_1) and (F_0, F_1) after transformation when comparing two structures C_0 and C_1.

Finally, we copy the coordinates from the selected protein to form the backbone conformation. Figure 8 shows the final conformation.

4 Experiment Results

In this section, we will show our experimental results and analyze the performance of our algorithm. Our algorithm is implemented on PC with AMD Duron processor 1000 MHZ and 256 MB RAM. 90% sequence similarity means that given two sequences, one can be replaced by randomly inserting, deleting or replacing 10% characters on the other sequence. Our test data include sequences with 30% similarity through 100% similarity of real biological sequences.

We compare the results of the protein prediction methods with folding algorithms and the algorithms without folding algorithms. There are two sequences in each test case. One is the template sequence, denoted as A, with a known structure. The other is the target sequence, denoted as B, without knowing its structure. In our test data, we get the protein structure files from PDB (http://www.rcsb.org/pdb/index.html). Then we use these proteins to be the structure-unknown ones and compare the experimental results with their original real structures from the PDB file. Table 6 shows the experimental results of comparing protein 5CPV and other protein sequences with different sequence similarity.

In our experiment, the prediction method with folding algorithm is referred to our algorithm in the previous section. The method without folding algorithm is described as follows:

Step 1. Search homologous proteins and align the sequences.

Step 2. Identify structurally conserved and structurally variable regions. Generate coordinates from template residues to the target protein.

Step 3. Arrange those region that are not structurally conserved from database. If the template we found are 100% sequence similarity related, then copy these coordinates from the first template.

Structure similarity represents the RMSD value of the real conformation of the target and template. ORMSD is the RMSD value of the predicting conformation formed without folding algorithm and the real target conformation. The C_score we use in Table 6 is the percentage of the structure similarity, which is defined as

$$C\_score = \frac{the\ CFAS}{the\ upper\ bound\ of\ the\ optimal\ CFAS}$$

where *CFAS* means the curve fitting alignment score.

If we get an optimal structure alignment, the upper bound of the score is $2 \times (l-3)$, where l is the length of the target protein sequence.

Take protein sequence 5CPV for example. The length of the protein sequence is 109. So the upper bound of the optimal curve fitting alignment score would be $(109-3) \times 2 = 212$. Thus if we the get a perfect template structure, the C_score of this optimal structure will be 1.

For structure similarity and ORMSD of cases 1-3 in Table 6, there is no improvement because of the high

Table 6: The experimental results of protein prediction algorithms with folding and without folding methods. Structure similarity: the RMSD value of the real conformation of the target and template. C: ORMSD(Å) the RMSD value of the conformation formed without folding algorithm. D: FRMSD(Å), the RMSD value of the conformation formed with folding algorithm. E: CNF_value, the C_score of the conformation formed without folding algorithm. F: CWF_value, the C_score of the conformation formed with folding algorithm.

		Target	Template	Length	Sequence similarity	Structure similarity	C	D	E	F
1		5CPV	1CDP	109	100.0%	0.2	0.2	0.2	1.000	1.000
2		5CPV	1B8C	108	97.2%	1.4	1.4	1.4	0.978	0.978
3		5CPV	1BU3	109	82.7%	0.6	0.6	0.6	0.901	0.915
4		5CPV	1A75	108	80.2%	1.0	0.8	1.0	0.842	0.889
5		5CPV	2PVB	107	76.4%	0.6	0.6	1.0	0.892	0.909
6		5CPV	2PAS	110	61.7%	1.2	1.2	1.2	0.62	0.778
7	v	5CPV	5PAL	109	50.5%	1.7	1.6	1.2	0.534	0.872
8	v	5CPV	1C7V	81	33.9%	3.3	3.2	2.4	0.487	0.512
9	v	5CPV	1C7W	81	33.8%	3.4	3.2	2.4	0.345	0.519
10	v	5CPV	1BOD	74	31.2%	3.9	3.8	2.4	0.21	0.504

sequence similarity between the template and the target proteins. In other cases, the algorithm with folding method gets some improvement.

In table 6, we can find that in cases 7, 8, 9 and 10, the prediction method with folding algorithms preforms better than those without folding algorithms. These protein sequences have lower sequence similarity and higher mismatching rate, that is, these folding sequences are long enough to be applied by the folding algorithm. Besides, if two sequences are very similar with the matched amino acids, the algorithm with folding do not have obvious improvement, such as cases 1, 2 and 3. In other cases, the algorithm with folding does not perform well. These cases have lower sequence similarity but the folding sequences are too short to perform the effects of the folding algorithms.

In Table 6, template protein sequences with high sequence similarity seem to get a better solution. See column sequence similarity and structural similarity. However, Table 7 illustrates the other condition that supports our theory. High sequence similarity does not always represent high structural similarity. Note that for cases 6-10, the template protein sequences with lower sequence similarity works better than those with higher sequence similarity. For example, the sequence similarity of case 1 is the highest one but the structural similarity is the worst one. That is, template protein sequences with higher sequence similarity would not always get a better solution. So, we should not choose the template only with high sequence similarity when predicting the target protein.

Similarly, the prediction method with folding algorithm also preforms well when these protein sequences have lower sequence similarity in Table 7, such as cases 6-10. Among all cases, we can observe that case 9 gets the best solution. That means, when predicting the structure of protein 1LIN, templates with high sequence similarity are not the best choices and if we predict protein structure with lower similarity template, our protein structure prediction method with folding algorithm works better than those are not.

Note that the RMSD value means the distance difference between two proteins and the C_score means the percentage of the structure similarity. We can see that our protein prediction methods with folding algorithms performs better than that without folding algorithms in C_score. It is because that the folding algorithm chooses the structures with similar shape.

Table 8 shows the experimental results of several RNase A amino acid sequences. In previous researches, most RNase A's show about 50% sequence homology. Since our prediction method with folding algorithm performs well on the protein sequences with lower similarity, we test these RNase A protein sequences. In our test, when the similarity is below a threshold, the folding algorithm actually performs well.

We can also apply the curve fitting structure alignment to measure the four folding algorithms. Given a protein with its structure, we are trying to use the various folding algorithms to predict its conformation and compare with its real structure. The experimental results are shown in Table 9. In cases 1 and 2, the proteins have the shorter length. We can see that the folding algorithm performs similarly except that the U-fold is worse. With the length of protein sequences getting

Table 7: The experimental results of protein prediction methods with folding algorithms and the algorithms without folding algorithms.C: ORMSD(Å) the RMSD value of the conformation formed without folding algorithm. D: FRMSD(Å), the RMSD value of the conformation formed with folding algorithm. E: CNF_value, the C_score of the conformation formed without folding algorithm. F: CWF_value, the C_score of the conformation formed with folding algorithm.

		Target	Template	Length	Sequence similarity	Structure similarity	C	D	E	F
1		1LIN	1CFD	148	100.0%	4.4	4.4	4.4	0.201	0.213
2		1LIN	2BBN	148	98.6%	3.5	3.5	3.5	0.346	0.368
3		1LIN	1IQ5	149	89.3%	3.7	3.6	3.5	0.343	0.369
4		1LIN	1CMF	73	75.0%	3.6	3.5	3.6	0.341	0.364
5		1LIN	1TNW	162	68.4%	2.8	2.6	2.8	0.438	0.556
6	v	1LIN	1EW7	161	66.7%	3.2	3.2	2.9	0.302	0.381
7	v	1LIN	1DTL	161	52.9%	2.0	1.9	1.6	0.532	0.753
8	v	1LIN	1CMG	73	44.8%	1.6	1.4	1.2	0.709	0.802
9	v	1LIN	1AVJ	161	32.9%	1.4	1.2	1.0	0.820	0.899
10	v	1LIN	1PVB	108	30.7%	3.2	2.9	2.3	0.496	0.504

Table 8: The experimental results of protein prediction methods with folding algorithms and the algorithms without folding algorithms in several RNase A amino acid sequences.

		Target	Length	Template	Length	Sequence similarity	CNF_value	CWF_value
1		1A2W	124	1HI5	134	38.0%	0.581	0.625
2		1ANG	123	1A2W	124	35.1%	0.63	0.693
3		1DYT	133	1HI5	134	62.4%	0.671	0.679
4		1DYT	133	1ANG	123	31.1%	0.60	0.72
5		1RCN	124	1HI5	134	37.6%	0.43	0.671
6		1RCN	124	1ANG	123	33.3%	0.35	0.55
7		1F0V	124	1HI5	134	36.2%	0.21	0.42
8		1F0V	124	1ANG	123	32.0%	0.52	0.52
9		1RBW	124	1RNF	120	40.7%	0.51	0.56
10		1RBW	124	1HI5	134	37.6%	0.32	0.35

Table 9: Comparison of different folding methods.

	Protein	Length	Folding Type	Score
1	1IG5	75	U-fold	32
			C-fold	51
			S-fold	53
			GA-approach	52
2	1D3Z	76	U-fold	43
			C-fold	69
			S-fold	71
			GA-approach	75
3	4FXC	98	U-fold	54
			C-fold	85
			S-fold	77
			GA-approach	96
4	5CPV	109	U-fold	98
			C-fold	115
			S-fold	147
			GA-approach	165
5	1AZU	128	U-fold	109
			C-fold	121
			S-fold	157
			GA-approach	195

longer, the GA approach performs better than others, and the U-fold still gets the worst performance. As the length of proteins gets longer and longer, it is obvious that the GA approach gets better score.

5 Conclusion

In this paper, we first give a brief survey on the algorithms for the protein structure prediction problem. Then, we propose a heuristic algorithm to predict the structure of a protein. Our algorithm consists of the homology modeling method and the folding algorithms. For comparing the folding model and the real protein structure, we use the curve fitting method as our structure alignment method. The curve alignment can also be used to evaluate the degree of similarity of two structures. By our experimental results, our protein structure prediction method performs well when the two protein sequences are not very similar.

The C_score we use in the structure alignment is a new measurement, which is different from the RMSD measurement. The RMSD computes the distances over all atoms, while the C_score measures the percentage of the structure similarity. And the curve alignment can also point out which region is "well-predicted".

In previous researches, the only criterion to determine the folding algorithm is *guaranteed performance ratio*. This criterion judges the quality of each folding algorithm by evaluating their energy but not the similarity of their structures. We propose a structure alignment method based on curve fitting that judges which conformation generated by various folding algorithms is closer to the true protein structure. Thus, the structure alignment with curve fitting is a useful method for evaluating the accuracy of the HP model conformation.

References

[1] R. Agarwala, S. Batzoglou, and V. Dancik, "Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP model," *Journal of Computational Biology*, Vol. 4, No. 3, pp. 275–296, 1997.

[2] T. Akutsu and H. Arimura, "On approximation algorithms for local multiple alignment," *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, Tokyo, Japan, pp. 1–7, 2000.

[3] B. Berger and T. Leight, "Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete.," *Journal of Computational Biology*, Vol. 5, No. 1, pp. 27–40, 1998.

[4] T. L. Blundell, B. L. Sibanda, M. J. E. Sternberg, and J. M. Thornton, "Knowledge-based prediction of protein structures and the design of novel molecules.," *Nature*, Vol. 326, pp. 347–352, 1987.

[5] T. Creighton, "The protein folding problem," *Science*, Vol. 240, pp. 267– 344, 1988.

[6] P. Crescenzi, D. Goldman, C. Capadimitriou, A. Piccolboni, and M. Yannakakis, "On the complexity of protein folding," *Journal of Computational Biology*, Vol. 5, No. 1, pp. 409–422, 1998.

[7] K. Dill, "Theory for the folding and stability of globular proteins," *Biochemistry*, Vol. 24, p. 1501, 1985.

[8] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design : A Practical Guide*. Boston: Academic Press, second ed., 1990.

[9] A. Fraenkel, "Complexity of protein folding," *Bulletin of Mathematical Biology*, pp. 1199–1210, 1993.

[10] C. F. Gerald and P. O. Wheatley, *Applied Numerical Analysis*. Addison Wesley Publishing, fourth ed., 1990.

[11] H. Hagen, *Curves and Surfaces Design*. SIAM Activity Group on Geometric Design, 1992.

[12] W. Hart and S. Istrail, "Fast protein folding in the hydrophobic-hydrophilic model within three-eights of optimal," *Journal of Computational Biology*, Vol. 3, No. 1, pp. 53–96, 1996.

[13] W. Hart and S. Istrail, "Lattice and off-lattice side chain models of protein folding: Linear time structure prediction better than 86% of optimal," *Journal of Computational Biology*, Vol. 4, No. 3, pp. 241–259, 1997.

[14] W. Hart and S. Istrail, "Robust proofs of NP-hardness for protein folding: general lattices and energy potentials," *Journal of Computational Biology*, Vol. 4, No. 1, pp. 1–22, 1997.

[15] L. Holm and C. Sander, "3-D lookup: fast protein structure database seaches at 90 reliability.," *Proceedings of 3rd International Conference on Intelligent Systems for Molecular Biology.*, Cambridge, UK., pp. 179–187, 1995.

[16] R. C. T. Lee, "Computational biology." http://www.csie.ncnu.edu.tw/, Department of Computer Science and Information Engineering, National Chi-Nan University, Taiwan, 2001.

[17] R. Lewin, "When does homology mean something else?," *Science*, Vol. 237, p. 1570, 1987.

[18] G. Mauri, A. Piccolboni, and G. Pavesi, "Approximation algorithms for protein folding prediction.," *Proceedings of the 10th Annual Symposium on Discrete Algorithms (SODA)*, San Antonio, USA, pp. 945–946, 1999.

[19] F. Richards, "The protein folding problem," *Scientific American*, Vol. 264, No. 1, pp. 54–63, 1991.

[20] A. Sali, E. Shakhnovich, and M. Karplus, "How does a protein fold?," *Nature*, Vol. 369, pp. 248–251, 1994.

[21] T. B. Sebastian, P. N. Kelin, and B. Kimia, "Alignment-based recognition of shape outlines.," *Proceedings of 4th International Workshop on Visual Form*, Capri, Italy, pp. 606–618, 2001.

[22] J. Setubal and J. Meidanis, *Introduction to Computational Molecular Biology*. PWS Publishing Company, Boston, second ed., 1997.

[23] N. Siew and D. Fischer, "Convergent evolution of protein structure prediction and computer chess tournaments:CASP, Kasparov, and CAFASP.," *IBM System Journal*, Vol. 40, No. 2, pp. 410–425, 2001.

[24] C. N. Storm and R. B. Lyngso, "Prediction of protein structures using simple exact models.." Technical Report. University of Aarhus, Denmark, 1996.

[25] W. R. Taylor and C. A. Orengo, "Protein structure alignment.," *Journal of Molecular Biology*, Vol. 208, pp. 1–22, 1989.

[26] R. Unger and J. Moult, "Finding the lowest free energy conformation of a protein is NP-hard problem: Proof and implications," *Bulletin of Mathematical Biology*, Vol. 55, No. 6, pp. 1183–1198, 1993.

[27] R. Unger and J. Moult, "Genetic algorithms for protein folding simulations," *Journal of Molecular Biology*, Vol. 231, No. 1, pp. 75–81, 1993.

[28] M. Waterman, *Introduction to Computational Biology: Maps, Sequences and Genomes*. Chapman and Hall, London: CRC Press, 1995.