An Ant Colony Optimization Approach for the Protein Side Chain Packing Problem [*]

Jing-Liang Hsin, Chang-Biau Yang[†], Kuo-Si Huang
Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung 804, TAIWAN
[†]cbyang@cse.nsysu.edu.tw

Chia-Ning Yang
Department of Medical Imaging and Radiological Sciences
I-Shou University
Kaohsiung County 840, TAIWAN
cnyang@isu.edu.tw

Abstract: The protein side chain packing problem (PSCPP) is an essential issue for predicting structure in proteomics. PSCPP has been proved to be NP-hard. In this paper, we propose a method for solving PSCPP by transforming it to the graph clique problem, and then applying the ant colony optimization (ACO) algorithm to solve it. We build the coordinate rotamer library based on the pair of dihedral angles of backbones to reduce the required time. To evaluate the goodness of a solution of the ACO algorithm, we use a simple score function with four factors: disulfide bonds, intermolecular hydrogen bonds, charge-charge interactions and van der Waals interactions. The experimental results show that our score function is biologically sensible. We compare our computational results with the results of SCWRL 3.0 and the residue-rotamer-reduction (R3) algorithm. The accuracy of our method outperforms both of them.

Key–Words: bioinformatics, protein structure, side chain, ACO (ant colony optimization), rotamer

1 Introduction

Side chain prediction plays an important role in investigating protein tertiary structure and function prediction. It has become a critical problem in many protein conformation prediction methods [1]. Usually, the *protein side chain packing problem* (PSCPP) assumes that the backbone of the target protein has been determined. For each residue of a polypeptide chain, there is a set of possible *rotamers*. The problem here is to choose one suitable rotamer for each residue such that the total energy of the protein is minimized. In other words, given a protein backbone conformation, PSCPP is to construct suitable side chains, which may be accomplished by comprehensively searching all possible rotamer conformations.

PSCPP can be transformed into a combinational optimization problem, and it has been shown to be NP-hard [7]. It implies that, in the worse case, any global optimization method for solving PSCPP is likely of exponential time complexity. Recently, some heuristic methods were proposed for solving PSCPP.

SCWRL 3.0 [2] is widely used to solve PSCPP. It uses an undirected graph to model this combinatorial problem. In the method, all possible rotamers of a side chain form a set of vertices. If two rotamers have nonzero interaction energy, they are connected by an edge in the graph. The resulting graph consists of many connected components, and it can be further broken into biconnected components. Note that in a biconnected component, there are at least two vertex disjoint paths between any pair of vertices. Now, the combinatorial problem is reduced to finding out the biconnected components with minimum energy and these results can be further combined to identify the global conformation with minimum energy. SCWRL 3.0 uses a simple energy function based on the backbone-dependent rotamer library and a linear repulsive steric energy.

The residue-rotamer-reduction (R3) algorithm [13] also solves PSCPP and it is faster than SCWRL 3.0. It applies graph theory to solve the combinatorial problem. R3 is similar to SCWRL 3.0 in rotamer library and energy function. The basic idea of R3 is to integrate residue reduction and rotamer reduction techniques.

In this paper, we shall propose a method for solving PSCPP. We first build a coordinate rotamer library as the template set, so we need not use the complicated energy function to calculate the bond length and bond angle. Our method is to reduce PSCPP to the graph clique finding problem and then we can apply the ant colony optimization (ACO) approach to search for suitable rotamers in the rotamer library. To measure the goodness of a solution in the ACO approach, we define a simple score function, which involves four factors: disulfide bonds, intermolecular hydrogen

[*] This research work was partially supported by the National Science Council of Taiwan under NSC-95-2221-E-110-102.

bonds, charge-charge interactions and van der Waals interactions. We perform experiments on two test sets, which are extracted from Xiang and Honig [12] and Canutescu et al. [2], respectively. The experimental results show that our score function is biologically sensible. And, the accuracy of our method outperforms both SCWRL 3.0 [2] and the R3 algorithm [13].

2 Preliminaries

Suppose there are four atoms A-B-C-D. The *torsion angle* of bond B-C is described by the angle between the plane containing atoms A, B and C, and the plane containing atoms B, C and D. This angle is also called the *dihedral angle*. The position of degree 0 of the torsion angle is given by the conformation in which the projections of A-B and C-D coincide (*cis*). Two isomeric conformations of four atoms A-B-C-D are shown in Figure 1. The torsion angle is defined to be positive or negative if bond A-B may coincide with bond C-D when bond A-B is rotated clockwise or counterclockwise, respectively. The torsion angle is measured in the range from degree -180 to 180, rather than from degree 0 to 360. It can be calculated by the 3-D coordinates of the four points. The norms to the two planes $\vec{N_1}$ (the plane containing atoms A, B and C), $\vec{N_2}$ (the plane containing atoms B, C and D), and dihedral angle θ are given as follows.

$$\vec{N_1} = \frac{\vec{AB} \times \vec{BC}}{|\vec{AB} \times \vec{BC}|}; \vec{N_2} = \frac{\vec{BC} \times \vec{CD}}{|\vec{BC} \times \vec{CD}|} \quad (1)$$

$$\theta = \arccos \frac{\vec{N_1} \cdot \vec{N_2}}{|N_1||N_2|} \quad (2)$$

Note that the dihedral angle is positive or negative can be decided by the following rules:

$$s = \vec{BC} \cdot (\vec{N_2} \times \vec{N_1}) \quad (3)$$

$$\theta' = \begin{cases} \pi - \theta & s \geq 0 \\ -(\pi - \theta) & s < 0 \end{cases} \quad (4)$$

The general structure of an amino acid consists of an amino group, a carboxyl group and an R group. The R group is a side chain bound to the α-carbon. Two or more amino acids can be bound together by peptide bonds, which are formed between the carboxyl group of one amino acid and the amino group of the next one.

The backbone conformation is described by two dihedral angles, ϕ and ψ, per residue. The side chain also has dihedral angles [6], as shown in Figure 2. Within each amino acid residue, there are two bonds with reasonably free rotation. One is the bond between the α-carbon and the nitrogen and the other is the bond between the α-carbon and the carboxyl carbon. The combination of the planar peptide group and the two freely rotating bonds has important implications for the three-dimensional conformations. In residue i, angles ϕ_i and ψ_i are defined by the atoms of C_{i-1}-N_i-C_i^α-C_i and N_i-C_i^α-C_i-N_{i+1}, respectively. The torsion angle X_{1i} of side chain is defined by the atoms of N_i-C_i^α-C_i^β-X_i [6]. Here, X is the atom with the highest priority if more than one atom binds to C^β. The rule of priority determination is as follows. The atom with higher atomic number has higher priority. If two atoms bound to C^β are the same, the ligands bound to these two atoms are used to determine the priority. A double bond has higher priority than a single bond. If two branches are the same, the one with the smaller absolute value has higher priority. If two branches have the same torsion angle degrees 90 and -90, respectively, the former one is chosen.

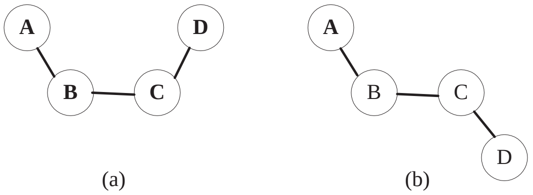

Figure 1: Two isomeric conformations of A-B-C-D. (a) *cis* conformation. (b) *trans* conformation.

3 Our Method

3.1 Our Model of PSCPP

Let $R = \{V_1, V_2, \ldots, V_n\}$ be the set of residues of the target protein whose side chain conformation is desired to be predicted. Each residue in R contains one or more possible rotamers. We transform PSCPP into a graph model as follows.

Node $v_{i,j}$ represents the jth possible rotamer of the ith residue. For two nodes (rotamers) $(v_{i,j}$ and $v_{m,n})$ associated with two residues V_i and V_m, there is an edge $(v_{i,j}, v_{m,n})$ connecting them if these two nodes do not collide with each other and $i \neq m$. Note that there is no edge connecting any two nodes $v_{i,j}$ and $v_{i,k}$, $j \neq k$, since they are associated with the same residue and only one rotamer for each residue can be selected.

Now, we can use an undirected graph $G = (V, E)$ to represent the rotamers of the side chain in a protein. Let $V_i = \{v_{i,j} |\ v_{i,j}$ does not collide with the backbone atoms$\}$. Then we have $V = \bigcup V_i$ and $E = \{(v_{i,j}, v_{m,n}) |\ v_{i,j}$ does not collide with $v_{m,n}$ and

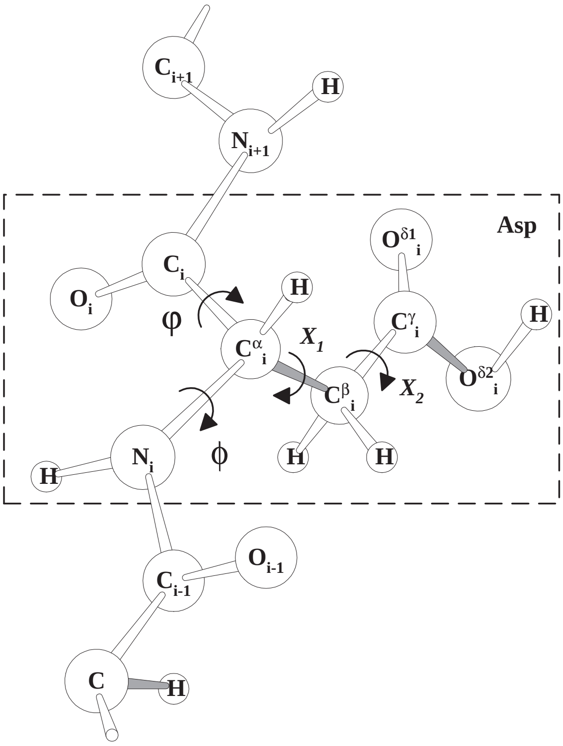

Figure 2: The dihedral angles in a polypeptide chain.

$i \neq m$ }. PSCPP becomes to find a clique (complete subgraph) Q of G, where $|Q| = |R|$. Here, one clique represents one feasible solution of PSCPP. Since there may be more than one such clique, we can invoke our score function to evaluate the goodness of each solution.

The ant colony optimization (ACO) algorithm was presented by Dorigo [5]. The ACO algorithm imitates the behavior of real ants, and it has been successfully applied to solve NP-complete problems, such as the *traveling salesperson problem* (TSP) [4]. In our ACO approach for PSCPP, the probability $p_k(s, u)$ for selecting an edge and the pheromone update formula $\tau_{s,u}(t+1)$ are given as follows.

$$p_k(s,u) = \frac{[\tau_{s,u}(t)]^{\alpha}[\eta(u)]^{\beta}}{\sum\limits_{w \in V_{i+1}} [\tau_{s,w}(t)]^{\alpha}[\eta(w)]^{\beta}}, \qquad (5)$$

$$\tau_{s,u}(t+1) = (1-\rho)\tau_{s,u}(t) + \sum_{k=1}^{m_{s,u}} \Delta\tau^{k}_{s,u}(t), \qquad (6)$$

where $s \in V_i$ and $u \in V_{i+1}$, $\tau_{s,u}(t)$ represents the concentration of pheromone on the edge from rotamer s to rotamer u in generation t, and $\eta(u)$ is the existent probability of rotamer u. The overall steps of our ACO algorithm for PSCPP are given as follows:

Algorithm ACO Algorithm for PSCPP

Input: The set of backbone coordinates of one protein and the rotamer library.

Output: The clique with near minimum score.

Step 1: Build the graph representation of PSCPP for the input.

Step 2: Set parameters and initialize pheromone trails.

Step 3: Each ant k chooses one rotamer u of each residue i according to Formula 5, for all $1 \leq i \leq n$. These rotamers of all residues form a clique.

Step 4: Update the pheromone trails with Formula 6.

Step 5: If the current best solution has not exceeded some percent after some predefined generations or the number of generations has reached the predefined value, return the clique with the minimum score; otherwise, go to Step 3.

The input data includes the three-dimensional coordinates of all atoms of the backbone. In Step 1, we build the graph model for representing PSCPP. We load the input data and then calculate the dihedral angles ϕ and ψ of each residue. According to the pair of dihedral angles, we get rotamers which do not collide with backbone from the rotamer library. The rotamer information contains the three-dimensional coordinate of each atom and existent probability of each rotamer. Step 2 sets the parameters, such as the weights α and β, the population and generation of the ant, the initial value of pheromone and the rate of pheromone evaporation. In Step 3, the probability formula includes two terms, the concentration of pheromone and the existent probability of a rotamer. Each ant k travels from the first residue to the last one, and chooses a rotamer for each residue from the rotamer set. The ant can not choose a rotamer that collides with any rotamer chosen before. In other words, the ant has to choose the node which is connected to each previously chosen nodes. Thus, a clique is formed after Step 3. In Step 4, we calculate the score of each solution (clique) by the score function. The score value is regarded as the concentration of pheromone that the ant leaves on her clique. The output of the algorithm is the clique with near minimum score. With the clique, backbone and rotamer coordinates, we can produce the PDB format file of the target protein.

3.2 The Rotamer Library

In the protein tertiary structure, the firm peptide bonds restrain the dihedral angles flexibility appreciably.

Table 1: The number of rotamers for each amino acid in our rotamer library.

	A.A.	# of Rotamers		A.A.	# of Rotamers
1	Ala	30830	11	Asn	16069
2	Cys	5150	12	Pro	16601
3	Asp	21256	13	Gln	13392
4	Glu	23778	14	Arg	16649
5	Phe	14652	15	Ser	21014
6	His	8016	16	Thr	20546
7	IlE	20458	17	Val	26076
8	Lys	21599	18	Trp	5220
9	Leu	31430	19	Tyr	12641
10	Met	7136			
Total					332513

The previous studies showed that each amino acid residue can take a three-dimensional position from a finite set of statistically significant conformations known as rotamers [8, 10].

Rotamer libraries can be classified into backbone-dependent and backbone-independent ones [9, 10]. Backbone-dependent rotamer libraries include information on side chain dihedral angles and backbone dihedral angles ϕ and ψ. Whereas backbone-independent libraries ignore such dependence. Our rotamer library is backbone-dependent, and it stores all pairs of dihedral angles ϕ and ψ. When predicting the side chain of a residue, we can first calculate its dihedral angles ϕ and ψ. Then, to find possible candidates from the rotamer library, we choose the subset of rotamers whose dihedral angles are close to ϕ and ψ. With this way, we can decrease the searching space of possible rotamers and improve the efficiency of the ACO algorithm.

The accuracy of side chain prediction depends primarily on the quality of the rotamer library [12]. Our coordinate rotamer library reserves the bond lengths and bond angles, which are kept in a standard rotamer library. The raw data is based on a set of 850 proteins which were used in the backbone-dependent rotamer library proposed by Dunbrack and Karplus [10]. We calculate dihedral angles of the backbone, and get the three-dimensional coordinate of each atom of side chain. Our rotamer library is divided into 19 files, each corresponding to one amino acid. Note that there is no file for glycine (Gly) since it has only one hydrogen atom on its side chain. The R group of alanine is a methyl group. Even though alanine and glycine are not included in the calculation of prediction accuracy, we still create the rotamer library of alanine, because we need the rotamer library to calculate the coordinate of C$^{\beta}$ of alanine. Table 1 lists the number of possible rotamers of each amino acid in our rotamer library.

```
N  CA  C  O  CB  OG
SER     -55   -43     -58  -1.440   0.498    0.130  -2.343   -0.343  -0.551
SER     -55   -43     -68   0.596   0.558    1.292   2.004    0.376   1.304
SER     -55   -44      66  -1.233  -0.745  -0.509   -1.001  -1.361  -1.761
SER     -55   -44      58   1.299  -0.444  -0.712    1.120  -1.398  -1.733
SER     -55   -44      54  -1.054  -1.094  -0.252  -0.742   -2.303   0.433
SER     -55   -44      54  -0.926   0.859  -0.871  -2.259    0.842  -0.386
SER     -55   -44     -60  -1.156  -1.001  -0.100  -0.728  -2.260    0.351
SER     -55   -44     -60  -0.213   0.859    1.245   0.807   0.554    2.185
SER     -55   -44     -66   0.156   0.037  -1.519  -0.386  -1.149  -2.067
SER     -55   -44     -68   0.232   0.644    1.369   1.220  -0.063   2.095
SER     -55   -44     -72   0.402  -1.197   0.856  -0.511  -2.267    0.685
SER     -55   -44    -74  -1.237  -0.331  -0.853  -0.832  -1.060  -1.980
SER     -55   -45    178   1.386    0.040  -0.644    2.303   0.734    0.188
SER     -55   -45    172  -0.832   0.553  -1.157  -0.268    1.713  -1.642
SER     -55   -45     -56   0.361  -1.444   0.363   0.874  -1.469   1.680
SER     -55   -45   -178  -0.652   0.027   1.389  -0.306    1.219   2.072
SER     -55   -45   -178   0.501   1.020   1.025  -0.546   1.419   1.890
```

Figure 3: An example of the rotamer library of serine.

As an example, a piece of our rotamer library of serine is shown in Figure 3. Line one of the file lists all residue atoms of serine and the remaining lines show the rotamer data. The columns of the rotamer data are of the following format:

$$[\text{A.A.}]\ [\phi]\ [\psi]\ [X_1]\ [\text{3-D Coordinate}]$$

The first column is the code name of amino acid, and the second, third and fourth columns are dihedral angles ϕ, ψ and X_1, respectively. The angle values are rounded after the decimal point up or down. We set the interval range of the dihedral angle X_1 to be two degrees. The existent probability in our ACO algorithm is calculated by dividing the number of X_1's within the two degrees by the total rotamers that belong to the same pair of the dihedral angles ϕ and ψ. The field ”3-D Coordinate” means the three-dimensional coordinates of all side chain atoms. In Figure 3, columns five to seven are the x, y, and z coordinate values of the first atom on the side chain, respectively.

3.3 The Score Function in ACO

In our ACO algorithm, after the ants finish the travel of all residues, we should discriminate good solutions from bad solutions by a score function. The solution with higher score is the better one. And the score value is the parameter to evaluate the survivor of pheromone on the path. Our score function E considers some factors in protein tertiary structures, including disulfide bonds, intermolecular hydrogen bonds, charge-charge interactions, and van der Waals interactions as follows.

$$
\begin{aligned}
S_1 =& BonS \times \#(\text{disulfide bonds}),\\
S_2 =& BonH \times \#(\text{hydrogen bonds}),\\
S_3 =& BonC \times (\#(\text{different charge pairs})\\
& - \#(\text{same charge pairs})),\\
S_4 =& BonV \times \sum_{i,j} E_{i,j},\\
E =& S_1 + S_2 + S_3 + S_4 \qquad (7)
\end{aligned}
$$

In Formula 7, $BonS$, $BonH$, $BonC$, and $BonV$ indicate the bonuses or weights. In the simulation experiments of this paper, parameters $BonS$, $BonH$, $BonC$ and $BonV$ are set as $0.5S_4$, 5.0, 2.0 and 1.0, respectively. The simple linear repulsive energy function [1] is adopted in S_4 as shown in Formula 8, where $R_{i,j}$ is the van der Waals radius [11] for atoms i and j, and r is the interatomic distance for each atom pair.

$$
E_{i,j} = \begin{cases} 0 & r > R_{i,j} \\ 10 & r < 0.8254R_{i,j} \\ 57.273(1 - \frac{r}{R_{i,j}}) & \text{otherwise} \end{cases} \qquad (8)
$$

4 Experimental Results

We execute our algorithm on PC with AMD Athlon$^{\text{TM}}$ processor 1700 MHZ and 512 MB RAM. The operating system is Windows 2000 Professional. The input data includes the amino acid sequence and the three-dimensional coordinate of each atom of the backbone. The output data is a PDB format file. Our method is tested on two sets of proteins from the literature. The first test set of 25 proteins comes from Xiang and Honig [12]. Some proteins of the test set have several chains, but we only extract the residues in chain A. The second test set includes five proteins, which comes from Canutescu et al. [2]. The proteins of the second test set are harder than those of the first set. We install SCWRL 3.0 [2] on the same machine to run the two test sets. In addition, R3 is an online server [13]. We submit each protein of the two test sets, and then download the results from the R3 server.

Parameters α and β are two parameters to control the influence of the pheromone and the existent probability, respectively. To weight the effect of rotamer existent probability, we set $\beta = 0.5$. The amount of ants (ant population) is 50, which means that 50 ants work in one generation of the algorithm. Initial pheromone is assigned to be 1.0.

Table 2 compares the experimental results of our method, SCWRL 3.0 and R3 for the two test sets. For each test protein, column Target Protein shows its PDB code and length, where the length is the number of residues other than alanine and glycine, because the R groups of the glycine and alanine are a single hydrogen atom and a methyl group, respectively. The accuracy of the predictions is judged by the dihedral angle X_1. If the difference of predicted X_1 and its angle in the real structure is within 20 degrees, the predicted angle would be regarded as correct. Columns three, four and five in Table 2 show the percentages of correct prediction for X_1 by our method, SCWRL 3.0 and R3, respectively. In columns four and five, the values on the left side of slashes are the accuracies that we judge according to the IUPAC-IUB rules [6]. The values on the right side of the slashes are extracted directly from Xie and Sahinidis's results [13]. Our method for predicting X_1 has accuracy between 70.1% and 87.1% in the first test set. The accuracies of SCWRL 3.0 and R3 are from 64.0% to 84.7%, and from 54.0% to 81.7%, respectively. Table 2(b) shows the accuracy of our method ranges between 70.8% and 74.9%, and the SCWRL 3.0 and R3 gives the accuracy from 62.3% to 72.8% and from 61.5% to 66.6%, respectively. We can conclude that for the most of the test proteins, our method outperforms both SCWRL 3.0 and R3.

5 Conclusion

The knowledge-based method usually provides a better solution for protein side chain prediction than the ab initio method, because more and more protein structures have been revealed. However, how to choose a suitable template is still a big challenge. In this paper, we take ACO algorithm as our searching method. The score function and the rotamer library are two main factors for the ACO algorithm to search the globally optimal solution. Our simple score function is not a real energy function like AMBER [3] but it can distinguish good and bad solutions. The experimental results indicate that our score function is biologically sensible, and show that our method outperforms both SCWRL 3.0 and R3.

The rotamer library is the template set for searching. We build a backbone-dependent rotamer library with the three-dimensional coordinate of each side chain atom. The coordinate rotamer library conserves the bond length and bond angle of native structure. This leads our computational results to approach the native structure. The time spent by our method is much more than SCWRL 3.0 and R3, because the ACO algorithm spends much time to converge the nearly optimal solution. But the accuracy of our method is better than both SCWRL 3.0 and R3. We shall therefore concentrate on the improvement of the execution time of our program in the future.

References:

[1] M. J. Bower, F. E. Cohen, and R. L. Dunbrack Jr., "Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: A new homology modeling tool," *Journal of Molecular Biology*, Vol. 267, No. 5, 1997, pp. 1268-1282.

[2] A. A. Canutescu, A. A. Shelenkov, and R. L. Dunbrack Jr., "A graph-theory algorithm for rapid protein side-chain prediction," *Protein Science*, Vol. 12, 2003, pp. 2001-2004.

[3] D. A. Case and T. E. Cheatham, "The Amber biomolecular simulation programs," *Journal of Computational Chemistry*, Vol. 26, 2005, pp. 1668-1688.

[4] M. Dorigo and M. L. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, 1997, pp. 53-66.

[5] M. Dorigo, V. Maniezzo, and A. Colorni, *Ant system: An autocatalytic optimizing process*. Italy: Technical Report, Politecnico di Milan, 1991.

[6] IUPAC-IUB Commission on Biochemical Nomenclature, "Abbreviations and symbols for the description of the conformation of polypeptide chains," *Journal of Biological Chemistry*, Vol. 246, No. 24, 1970, pp. 6489-6497.

[7] N. A. Pierce and E. Winfree, "Protein design is NP-hard," *Protein Engineering Design and Selection*, Vol. 15, No. 10, 2002, pp. 779-782.

[8] J. W. Ponder and F. M. Richards, "Tertiary templetes for proteins: Use of packing criteria in the enumeration of allowed sequences for different structural classes," *Journal of Molecular Biology*, Vol. 193, 1987, pp. 775-792.

[9] R. L. Dunbrack Jr. and F. E. Cohen, "Bayesian statistical analysis of protein side-chain rotamer preferences," *Protein Science*, Vol. 6, 1997, pp. 1661-1681.

[10] R. L. Dunbrack Jr. and M. Karplus, "Backbone-dependent rotamer library for proteins: Application to side-chain prediction," *Journal of Molecular Biology*, Vol. 230, 1993, pp. 543-574.

[11] G. E. Schulz and R. H. Schirmer, *Principles of Protein Ptructure*. New York: Springer-Verlag, 1979.

[12] Z. Xiang and B. Honig, "Extending the accuracy limits of prediction for side-chain conformations," *Journal of Molecular Biology*, Vol. 311, 2001, pp. 421-430.

[13] W. Xie and N. V. Sahinidis, "Residue-rotamer-reduction algorithm for the protein side-chain conformation problem," *Bioinformatics*, Vol. 22, No. 2, 2006, pp. 188-194.

Table 2: Comparison of our method with SCWRL 3.0 and R3. (a) The first test set. (b) The second test set. Columns 3∼5 show the X_1 prediction accuracy in percentage.

Target Protein		Our Method	SCWRL 3.0	R3 Method
Protein	Length	X_1	X_1	X_1
1AAC	85	87.1	84.7/95	76.5/86
1AHO	54	85.2	68.5/67	64.8/65
1B9O	112	70.5	68.8/73	66.1/77
1C5E	71	81.7	81.7/86	73.2/82
1C9O	53	84.9	66.0/72	71.7/70
1CC7	66	80.3	68.2/83	63.6/79
1CEX	146	85.6	76.7/82	75.3/77
1CKU	60	81.7	76.7/82	68.3/80
1CTJ	61	77.0	68.9/79	70.5/80
1CZ9	111	70.3	64.0/73	64.0/76
1CZP	83	79.5	77.1/86	73.5/81
1D4T	89	77.5	76.4/86	67.4/82
1IGD	50	82.0	68.0/74	54.0/68
1MFM	118	75.4	68.6/80	70.3/81
1PLC	82	72.0	67.1/72	70.7/71
1QJ4	221	71.5	72.9/84	67.9/80
1QQ4	143	83.9	73.4/78	71.3/78
1QTN	134	86.6	74.6/82	67.9/78
1QU9	99	79.8	71.7/81	73.7/78
1RCF	142	79.6	83.8/86	81.7/80
1VFY	63	79.4	69.8/76	71.4/75
2PTH	151	82.1	78.8/83	78.1/84
3LZT	105	73.3	78.1/86	69.5/82
5P2L	144	78.5	70.8/78	63.2/71
7RSA	109	75.2	65.1/75	61.5/67

(a)

Target Protein		Our Method	SCWRL 3.0	R3 Method
Protein	Length	X_1	X_1	X_1
1A8I	704	73.4	71.3/80	64.1/75
1B0P	978	70.8	62.3/69	- /66
1BU7	399	74.9	70.4/78	64.4/72
1GAI	386	73.6	72.8/81	66.6/72
1XWL	496	71.5	66.7/73	61.5/72

(b)