Protein Secondary Structure Prediction Using Dynamic Programming

Jing ZHAO[1,2,3], Pei-Ming SONG[1], Qing FANG[1], and Jian-Hua LUO[1,2]*

[1]*School of Life Science & Technology, Shanghai Jiaotong University, Shanghai 200240, China;*
[2]*Shanghai Center for Bioinformation and Technology, Shanghai 200235, China;*
[3]*Logistical Engineering University, Chongqing 400016, China*

Abstract In the present paper, we describe how a directed graph was constructed and then searched for the optimum path using a dynamic programming approach, based on the secondary structure propensity of the protein short sequence derived from a training data set. The protein secondary structure was thus predicted in this way. The average three-state accuracy of the algorithm used was 76.70%.

Key words directed graph; dynamic programming approach; protein secondary structure

Protein structure prediction helps to facilitate our understanding of the protein function. It is commonly recognized that the 3-D structure of a protein can be accurately predicted when the prediction accuracy of the secondary structure reaches 80.00%. The prediction of the secondary structure using the primary structure is the main obstacle when predicting the 3-D structure of a protein. When predicting the secondary structure, the three-state accuracy Q_3 is used as a criterion to assess the prediction accuracy,

$$Q_3 = \frac{N_\alpha + N_\beta + N_c}{N} \qquad (1)$$

where N_α, N_β and N_c are respectively the number of residues in α-helix, β-sheet and other types predicted correctly, and N is the total number of amino acid residues predicted.

A number of computational methods have been developed for predicting the protein secondary structure, such as information theory methods, the nearest-neighbor method and the artificial neural network method. Information theory methods are based on the statistical characteristics of a single amino acid's propensity for a given conformational state. Examples of such methods include GOR1 [1], GOR3 [2], GORIV [3], and DSC [4]. Their Q_3 values are about 69.50%. Zvelebil *et al.* [5] used the alignment of homologous sequences and got a Q_3 value of 66.00%. The nearest-neighbor method is based on the conformational states of the best matches or nearest neighbors. An example of such a method is PREDATOR [6], whose Q_3 is about 68.00%. Yi and Lander's algorithm [8], NNSSP [9] and PHD [10] are based on artificial neural networks. The highest Q_3 of these artificial neural network methods is 74.00%. Recently, Ward *et al.* [11] used support vector machines and got a Q_3 value of 77.07%. Recent researches in this area have been mainly focused on the incorporation of existing methods to improve the prediction accuracy.

In the present paper, we introduce a novel method for the secondary structure prediction of a protein.

Analysis of Short Peptide Propensity

First, we downloaded all the 24,310 protein sequences and their secondary structure parameters from the DSSP (Database of Secondary Structure in Proteins, http://www.sander.ebi.ac.uk/dssp/) and NLR-3D [the Sequence-structure Database produced from sequence and annotation information extracted from three-dimensional structures in the Protein Databank (PDB), http://pir.georgetown.edu/pirwww/dbinfo/nrl3d.html]. Then we deleted the redundant and inferior sequences according to the following rules: (1) omit the homologous sequences; (2) delete those sequences with the wrong secondary structure notation; and (3) delete those sequences designated to be of low

Received: October 18, 2004 Accepted: January 17, 2005

This work was supported by the grants from the National High Technology Research and Development Program of China (No. 2002AA234021), the National Basic Research Program of China (No. 2002 CB 512800), the Science and Technology Committee of Shanghai (No. K02D 105501), and the Corporation Project of Shanghai Jiaotong University and Shanghai Second Medical University

*Corresponding author: Tel, 86-21-54742963; E-mail, jhluo@sjtu.edu.cn

quality by PROCHECK which checks the stereochemical quality of a protein structure (http://www.biochem.ucl.ac.uk/~roman/procheck/procheck.html) [6]. This process resulted in a set of 5100 sequences with high quality annotation in the secondary structure that was used for the short peptide propensity analysis.

The short peptides were divided into 10 secondary structure classes: (1) connecting peptides between α-helix and α-helix, denoted by αα; (2) connecting peptides between α-helix and β-sheet, denoted by αβ; (3) connecting peptides between β-sheet and β-sheet, denoted by ββ; (4) connecting peptides between β-sheet and α-helix, denoted by βα; (5) beginning peptides of α-helix, denoted by α_B; (6) beginning peptides of β-sheet, denoted by β_B; (7) terminal peptides of α-helix, denoted by α_E; (8) terminal peptides of β-sheet, denoted by β_E; (9) α-helical peptides, denoted by α; and (10) β-sheet peptides, denoted by β.

All the peptides of the 5100 sequences make up set Ω. Those peptides belonging to αα, αβ, ββ, βα, α_B, β_B, α_E, β_E, α and β, respectively, comprise subset Ω_i, i=0, 1,⋯, 9. The number of peptides in each secondary structure class is listed in **Table 1**.

Let $N(w,\Omega_i)$ be the occurrence frequency of peptide w in set Ω_i. The secondary structure propensity coefficient (SSPC) $P(w,\Omega_i)$ is then defined by,

$$P(w,\Omega_i)=\frac{N(w,\Omega_i)}{N(w,\Omega)},\ i=0,\ 1,\ \cdots,\ 9 \qquad (2)$$

The peptide conflict rate is defined as the percentage of peptides that belong to two or more secondary structure classes (αα, αβ, ββ, βα, α_B, β_B, α_E, β_E, α, β) in the total number of peptides. Through statistical analysis, we found that when the length of the peptide L is 4 amino acids, the conflict rate is too high and the secondary structure propensity is too low to be used in the prediction of the secondary structure. Statistical analysis results also show that L=5 amino acids is the best peptide length for the propensity analysis.

Construction of the Directed Graph

For a protein sequence $X=x_0x_1\cdots x_{N-1}$, where N is the length of the sequence, the short peptide $w[j]$ of length L for position j is defined as follows:

$$w[j]=x_jx_{j+1}\cdots x_{j+L-1} \qquad j=0,\ 1,\cdots,\ N-L$$

The SSPC of $w[j]$ is denoted by $P(w[j],\Omega_i)$ (i=0, 1,⋯, 9). The SSPCs of sequence X make up the matrix $P(X)$ of $10\times(N-L+1)$:

$$P(X)=\begin{bmatrix} P(w[0],\Omega_0) & P(w[1],\Omega_0) & \dots & P(w[N-L],\Omega_0) \\ P(w[0],\Omega_1) & P(w[1],\Omega_1) & \dots & P(w[N-L],\Omega_1) \\ \vdots & \vdots & \vdots & \vdots \\ P(w[0],\Omega_9) & P(w[1],\Omega_9) & \dots & P(w[N-L],\Omega_9) \end{bmatrix} \qquad (3)$$

When the propensity coefficient of αα $P(w[j],\Omega_0)>0$, the short peptide in position j is probably a connecting peptide of αα. This αα peptide is equivalent to a terminal peptide of α_E in position j–1 and a beginning peptide of α_B in position j+1. So the propensity coefficients of α_E in j–1 and α_B in j+1 are both equal to $P(w[j],\Omega_0)$. Therefore, for simplification, the propensity coefficients of αα can be included in the propensity coefficients of α_B and α_E by modifying the propensity coefficients of α_E and α_B as follows:

$$P(w[j-1],\Omega_6)=\max\{P(w[j],\Omega_0),P(w[j-1],\Omega_6)\}$$
$$P(w[j+1],\Omega_4)=\max\{P(w[j],\Omega_0),P(w[j+1],\Omega_4)\}$$

Similarly, the propensity coefficients of αβ can be included in the propensity coefficients of α_E and β_B by modifying the propensity coefficients of α_E and β_B as follows:

$$P(w[j-1],\Omega_6)=\max\{P(w[j],\Omega_1),P(w[j-1],\Omega_6)\}$$
$$P(w[j+1],\Omega_5)=\max\{P(w[j],\Omega_1),P(w[j+1],\Omega_5)\}$$

The propensity coefficients of ββ can be included in the propensity coefficients of β_E and β_B by modifying the propensity coefficients of α_E and β_B as follows:

$$P(w[j-1],\Omega_7)=\max\{P(w[j],\Omega_2),P(w[j-1],\Omega_7)\}$$
$$P(w[j+1],\Omega_5)=\max\{P(w[j],\Omega_2),P(w[j+1],\Omega_5)\}$$

The propensity coefficients of βα can be included in the propensity coefficients of β_E and α_B by modifying the propensity coefficients of α_E and β_B as follows:

$$P(w[j-1],\Omega_7)=\max\{P(w[j],\Omega_3),P(w[j-1],\Omega_7)\}$$
$$P(w[j+1],\Omega_4)=\max\{P(w[j],\Omega_3),P(w[j+1],\Omega_4)\}$$

Table 1 The number of peptides in each secondary structure class

Ω_0(αα)	Ω_1(αβ)	Ω_2(ββ)	Ω_3(βα)	$\Omega_4(\alpha_B)$	$\Omega_5(\beta_B)$	$\Omega_6(\alpha_E)$	$\Omega_7(\beta_E)$	Ω_8(α)	Ω_9(β)	Ω
10,785	15,247	16,397	15,950	28,843	35,409	28,802	35,423	797,536	282,152	1,666,006

The propensity coefficient $P(w[j],\Omega_4)$ of α_B in position *j* means that the probability of occurrence of the secondary structure α_B in position *j* is $P(w[j],\Omega_4)$. In addition, when the propensity coefficients of the α peptide in position *j*, *j*+1 and *j*+2 are high, the credibility of the secondary structure α_B in position *j* increases; otherwise, it decreases. Therefore, the credibility of α_B is defined as:

$$S(j,\alpha_B)=P(w[j],\Omega_4)[1+(P(w[j],\Omega_8)+P(w[j+1],\Omega_8)+P(w[j+2],\Omega_8))/3]$$

The credibility of β_B is defined in the same way:

$$S(j,\beta_B)=P(w[j],\Omega_5)[1+(P(w[j],\Omega_9)+P(w[j+1],\Omega_9)+P(w[j+2],\Omega_9))/3]$$

The propensity coefficient $P(w[j],\Omega_6)$ of α_E in position *j* means that the probability of occurrence of the secondary structure α_E in position *j* is $P(w[j],\Omega_6)$. Additionally, when the propensity coefficients of the α peptide in positions *j*, *j*–1 and *j*–2 are high, the credibility of the secondary structure α_E in position *j* increases; otherwise, it decreases. Therefore, the credibility of α_E is defined as:

$$S(j,\alpha_E)=P(w[j],\Omega_6)[1+(P(w[j],\Omega_8)+P(w[j-1],\Omega_8)+P(w[j-2],\Omega_8))/3]$$

and the credibility of β_E is defined similarly as:

$$S(j,\beta_E)=P(w[j],\Omega_7)[1+(P(w[j],\Omega_9)+P(w[j-1],\Omega_9)+P(w[j-2],\Omega_9))/3]$$

Therefore, the protein sequence $X=x_0x_1\cdots x_{N-1}$ corresponds to a matrix $S(X)$:

$$S(X)=\begin{bmatrix} S(0,\alpha_B) & S(1,\alpha_B) & \dots & S(N-L,\alpha_B) \\ S(0,\beta_B) & S(1,\beta_B) & \dots & S(N-L,\beta_B) \\ S(0,\alpha_E) & S(1,\alpha_E) & \dots & S(N-L,\alpha_E) \\ S(0,\beta_E) & S(1,\beta_E) & \dots & S(N-L,\beta_E) \end{bmatrix} \qquad (4)$$

Finally, a directed graph *G* is constructed from *S*(*X*) as follows.

(1) The vertex set {*node*(*j*), *j*=1, 2,⋯, *k*} is composed of *k* vertices.

A vertex in a directed graph *G* is defined as the linking region between two secondary structures. Its data structure is:

node(*j*){float α_E*score*, β_E*score*, α_B*score*, β_B*score*; int α_E*position*, β_E*position*, α_B*position*, β_B*position*, *position*}

where α_E*score*, β_E*score*, α_B*score* and β_B*score* are the respective credibility scores of α_E, β_E, α_B and β_B of *node*(*j*); α_E*position*, β_E*position*, α_B*position* and β_B*position* are the respective positions of α_E, β_E, α_B and β_B; and *position* is the position of the vertex *node*(*j*) in *X*. If there are *k* vertices in *S*(*X*), then the parameters of *node*(*j*), *j*=1, 2,⋯, *k* are calculated as follows:

$node(j).\alpha_E score=S(i_1^*,\alpha_E)$; $node(j).\ \alpha_E position=i_1^*$

where i_1^* satisfies:

$S(i_1^*,\alpha_E)=\max\{S(i,\alpha_E),position[j-1]<i<position[j]\}$

$node(j).\ \beta_E score=S(i_2^*,\beta_E)$; $node(j).\ \beta_E position=i_2^*$

where i_2^* satisfies:

$S(i_2^*,\beta_E)=\max\{S(i,\beta_E),position[j-1]<i<position[j]\}$

$node(j).\ \alpha_B score=S(i_3^*,\alpha_B)$; $node(j).\ \alpha_B position=i_3^*$

where i_3^* satisfies:

$S(i_3^*,\alpha_B)=\max\{S(i,\alpha_B),position[j]<i<position[j+1]\}$

$node(j).\ \beta_B score=S(i_4^*,\beta_B)$; $node(j).\ \beta_B position=i_4^*$

where i_4^* satisfies:

$S(i_4^*,\beta_B)=\max\{S(i,\beta_B),position[j]<i<position[j+1]\}$

where *j*=1, 2, ⋯, *k*, and assuming *position*[0]=0, *position*[*k*+1]=*N*–*L*.

(2) The weights of the directed arc from *node*(*i*) to *node*(*j*) represent the secondary structure propensity from *node*(*i*) to *node*(*j*) in *G*, where *i*<*j*. These are defined respectively as:

$$\omega_\alpha(i,j)=\frac{\sqrt{node(i).\alpha_B score\times node(j).\alpha_E score}}{j-i}, \qquad 1\le i<j\le k \qquad (5)$$

$$\omega_\beta(i,j)=\frac{\sqrt{node(i).\beta_B score\times node(j).\beta_E score}}{j-i}, \qquad 1\le i<j\le k \qquad (6)$$

where the denominator *j*–*i* represents the penalty for the leaping over the vertices between *node*(*j*) and *node*(*i*).

From the definitions above, the graph *G* with *k* nodes for secondary structure prediction is presented as the following matrix:

$$G=\begin{bmatrix} 0 & \omega_\alpha(1,2) & \omega_\alpha(1,3) & \omega_\alpha(1,4) & \cdots & \omega_\alpha(1,k) \\ \omega_\beta(1,2) & 0 & \omega_\alpha(2,3) & \omega_\alpha(2,4) & \cdots & \omega_\alpha(2,k) \\ \omega_\beta(1,3) & \omega_\beta(2,3) & 0 & \omega_\alpha(3,4) & \cdots & \omega_\alpha(3,k) \\ \omega_\beta(1,4) & \omega_\beta(2,4) & \omega_\beta(3,4) & 0 & \cdots & \omega_\alpha(4,k) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega_\beta(1,k) & \omega_\beta(2,k) & \omega_\beta(3,k) & \omega_\beta(4,k) & \cdots & 0 \end{bmatrix} \qquad (7)$$

Every element (from the second element onward) in the first row represents the weight of the α structure from the first vertex to other vertices, and every element (from the second element onward) in the first column represents the weight of the β structure from the first vertex to other vertices. Similarly, every element (from the third element onward) in the second row represents the weight of the α structure from the second vertex to other vertices, and every element (from the third element downward) in the second column represents the weight of the β structure from the second vertex to other vertices. The rest of the elements can be similarly explained. For example, a directed graph *G* with four vertices is shown in **Fig. 1**.

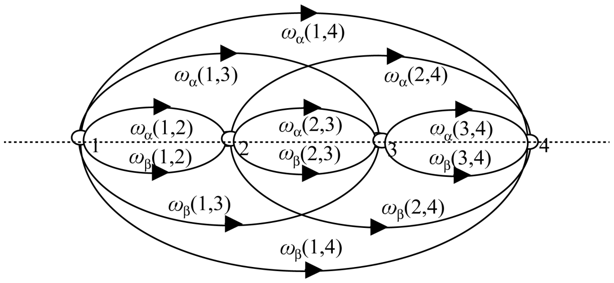

Fig. 1 The directed graph with four vertices for the protein secondary structure prediction

Searching for the Optimum Path

Utilizing the directed graph *G* with *k* vertices defined above, the secondary structure of the corresponding protein can be predicted. In graph *G*, a single directed path from the initial vertex to the terminal vertex represents one solution of the secondary structure prediction.

Let $E_x(i_p, i_{p+1})$ represent the directed arc from $node(i_p)$ to $node(i_{p+1})$ connected by structure x (x=α or β). The symbol "≻" is used to denote the junction between two arcs. Therefore, *Path*(*i,j*), which is the path composed of *z* arcs from *node*(*i*) to *node*(*j*), is denoted as:

$$Path(i, j)=E_x(i, i_1)\succ E_x(i_1, i_2)\succ E_x(i_2, i_3)\succ\cdots\succ E_x(i_{z-1}, j),\quad i<i_1<i_2<i_3<\cdots<i_{z-1}<i_z=j$$

The weight of the path is defined as:

$$\omega(Path(i, j))=[\omega_x(i, i_1)+\omega_x(i_1, i_2)+\cdots+\omega_x(i_{z-1}, j)]/z$$

This means the weight of *Path*(*i*, *j*) is defined as the average weights of all the directed arcs along this path. From this definition, the optimum path from the initial vertex to the terminal vertex of graph *G*—namely the path with the highest *w*(*Path*(*1*,*k*))—represents the optimum solution of the secondary structure prediction, corresponding to the solution with the highest mean credibility.

For graph *G* with *k* vertices, there exist $2^k k!$ paths from the initial vertex to the terminal vertex. The task of the protein secondary structure prediction becomes transformed into one of finding an optimum path, namely $Path^*$, which maximizes the $w(Path^*)$ in the $2^k k!$ paths. This task has exponential computational complexity. Such a challenge can be overcome efficiently by a dynamic programming approach (DPA).

Computing the optimum path by a dynamic programming approach is based on the optimum principle:

If $Path^*(1, i)$ is the optimum path from the initial vertex *node*(1) to the *i*th vertex of graph *G* and $E^*(i, j)$ is the optimum directed arc from *node*(*i*) to *node*(*j*), with $j\in\{i+1, i+2,\cdots, k\}$, then

$$Path^*(1,j)=Path^*(1,i)\succ E^*(i,j)$$

is the optimum path from the initial vertex *node*(1) to *node* (*j*).

Based on this principle, when computing the optimum path from the initial vertex *node*(1) to *node*(*j*), the optimum paths from *node*(1) to the senior vertices of *node*(*j*) should be computed in advance. Therefore, we may begin from the initial vertex *node*(1), and compute the optimum path from senior vertices to junior vertices.

Three parameters of *node*(*j*) are defined as follows:
$v(j)$: score of *node*(*j*), $v(j)=\max\{\omega(Path(1, j))\}$, representing the weight of the optimum path from the initial vertex to *node*(*j*);
$b(j)$: number of arcs in the optimum path from the initial vertex *node*(1) to *node*(*j*);
$U(j)$: ordered array composed of the vertex code and structure types, representing the optimum path from *node*(1) to *node*(*j*), where $j=1, 2,\cdots, k$, and $v(1)=0$, $b(1)=0$, $U(1)=\{1\}$.

Two parameters for the directed arc from *node*(*i*) to *node*(*j*) are defined as follows:
$\omega_x(i, j)$: weights of the optimum directed arc from *node*(*i*) to *node*(*j*); $\omega_x(i, j)=\max\{\omega_\alpha(i, j), \omega_\beta(i, j)\}$;
$B(i, j)$: structure types of the optimum directed arc from *node*(*i*) to *node*(*j*); if $\omega_x(i, j)=\omega_\alpha(i, j)$, then $B(i, j)=\alpha$, otherwise $B(i, j)=\beta$, where $1\le i<j\le k$.

The dynamic programming algorithm that searches for the optimum path is described in detail as follows:
Step 1: for $1\le i<j\le k$, calculate $\omega_x(i, j)$ and $B(i, j)$:

$\omega_x(i, j)=\max\{\omega_\alpha(i, j), \omega_\beta(i, j)\}$

if $\omega_x(i, j)=\omega_\alpha(i, j)$, then

$B(i, j)=\alpha$, otherwise $B(i, j)=\beta$.

Step 2: let $v(1)=0$, $b(1)=0$, $U(1)=\{1\}$;
Step 3: for each $j=2, 3,\cdots, k$, compute $v(j)$, $b(j)$ and $U(j)$ successively:

$$v(j)=\max\left\{\frac{v(p)\times b(p)+\omega_x(p,j)}{b(p)+1}\middle| p=1,2,\cdots,j-1\right\}$$

(1) If there exists only one p^* such that

$$v(j)=\frac{v(p^*)\times b(p^*)+\omega_x(p^*,j)}{b(p^*)+1}$$

then

$b(j)=b(p^*)+1$

Add $B(p^*, j)$ and j orderly to the end of set $U(p^*)$, and obtain $U(j)$.

(2) If there exist p_1, p_2, $\cdots$, p_k, such that $1\le p_1<p_2<\cdots<p_k\le j-1$, and

$$\frac{v(p_1)\times b(p_1)+\omega_x(p_1,j)}{b(p_1)+1}=\cdots=\frac{v(p_k)\times b(p_k)+\omega_x(p_k,j)}{b(p_k)+1}$$

$=v(j)$

then

$b(j)=b(p_k)+1$

Add $B(p_k, j)$ and j orderly to the end of set $U(p_2)$, and obtain $U(j)$ (the weight of the vicinity vertex is preferential).

Finally, we obtain $v(k)$, which is the weight of the optimum path from the initial vertex to the terminal vertex, and $U(k)$, which is the corresponding trial optimum path. For example, the optimum path in **Fig. 2** is 1-β-2-α-3-α-4, which means the β structure is predicted from *node*(1) to *node* (2), and the α structure is predicted from *node*(2) to *node* (4).

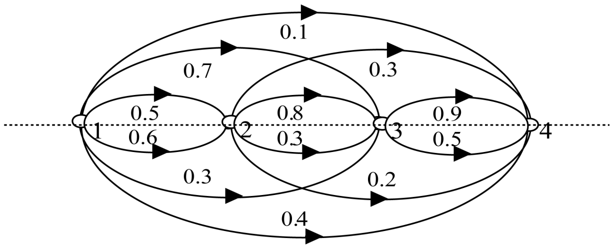

Fig. 2 Weighted direction graph with four nodes for protein secondary structure prediction

Results and Discussion

The data from 130 low-homologous proteins selected by Rost and Sander [10] were used to test the algorithm described here.

In Test 1, we divided the data into four groups, namely α, β, α+β & α/β and others, and predicted the protein secondary structure with GOR3, PHD and the DPA introduced in this paper. The Q_3 values of these methods are compared in **Table 2**. We can see that DPA performed better than GOR3 and PHD in almost every case in Test 1.

Test 2 was carried out to further investigate the performance of DPA. In Test 2, we partitioned the sequences into two subsets, Set I and Set II. Set I contains those sequences of which more than 90% are longer than 5 amino acids, while Set II contains the remaining sequences. The prediction results are listed in **Table 3**.

From **Table 2**, we can see that the performance of DPA

Table 2 Q_3 comparison of the prediction results in Test 1

Case	Sequence number	Residue number	GOR3 Q_3 (%)	PHD Q_3 (%)	DPA Q_3 (%)
All α	23	3247	64.7	83.10	80.53
All β	10	1092	48.6	73.96	75.60
α+β & α/β	40	9955	57.9	76.15	76.46
Others	57	10,143	57.7	72.67	75.93
Total	130	24,437	58.3	75.69	76.70

Table 3 Q_3 **comparison of the prediction results in Test 2**

Set	Sequence number	Residue number	GOR3 Q_3 (%)	PHD Q_3 (%)	DPA Q_3 (%)
I	28	3434	65.25	83.90	96.53
II	102	21,003	57.15	73.51	73.45
Total	130	24,437	58.30	75.69	76.70

for Set I is much better than that of GOR3 and PHD, while the results for Set II show no significant difference between DPA and PHD. This is because DPA's SSPC database omits those peptides whose lengths are less than 5 amino acids. Fortunately, Set II only contains a small number of sequences.

Conclusion

DPA can overcome the shortcomings of the methods based on a single amino acid's propensity because it utilizes SSPC, and it is faster because of the dynamic programming algorithm. DPA will perform even better if combined with other methods.

References

1 Garnier J, Osguthorpe DJ, Robson B. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. J Mol Biol, 1978, 120: 97–120

2 Gibrat JF, Robson B, Garnier J. Further developments of protein secondary structure prediction using information theory: New parameters and consideration of residue pairs. J Mol Biol, 1987, 198(3): 425–443

3 Garnier J, Gibrat JF, Robson B. GOR method for predicting protein secondary structure from amino acid sequence. Methods Enzymol, 1996, 266: 540–553

4 King RD, Sternberg MJE. Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. Protein Sci, 1996, 65(11): 2298–2310

5 Zvelebil MJ, Barton GJ, Taylor WR, Sternberg MJE. Prediction of protein secondary structure and active site using the alignment of homologous sequences. J Mol Biol, 1987, 195(4): 957–967

6 Frishman D, Argos P. Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence. Protein Eng, 1996, 9(2): 133–142

7 Frishman D, Argos P. Seventy-five percent accuracy in protein secondary structure prediction. Proteins, 1997, 27(3): 329–335

8 Yi TM, Lander S. Protein secondary structure prediction using nearest-neighbor methods. J Mol Biol, 1993, 232(4): 1117–1129

9 Salamov AA, Solovyev VV. Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignment. J Mol Biol, 1995, 247(1): 11–15

10 Rost B, Sander C. Prediction of protein secondary structure at better than 70% accuracy. J Mol Biol, 1993, 232(2): 584–599

11 Ward JJ, McGuffin LJ, Buxton BF, Jones DT. Secondary structure prediction with support vector machines. Bioinformatics, 2003, 19(13): 1650–1655

Edited by
Da-Fu DING