PROTEINS
STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

Threading without optimizing weighting factors for scoring function

Yifeng David Yang,[1] Changsoon Park,[2] and Daisuke Kihara[1,3,4*]

[1] Department of Biological Sciences, College of Science, Purdue University, West Lafayette, Indiana 47907
[2] Department of Statistics, College of Natural Science, Chung-Ang University, Seoul, Korea
[3] Department of Computer Science, College of Science, Purdue University, West Lafayette, Indiana 47907
[4] Markey Center for Structural Biology, College of Science, Purdue University, West Lafayette, Indiana 47907

ABSTRACT

Optimizing weighting factors for a linear combination of terms in a scoring function is a crucial step for success in developing a threading algorithm. Usually weighting factors are optimized to yield the highest success rate on a training dataset, and the determined constant values for the weighting factors are used for any target sequence. Here we explore completely different approaches to handle weighting factors for a scoring function of threading. Throughout this study we use a model system of gapless threading using a scoring function with two terms combined by a weighting factor, a main chain angle potential and a residue contact potential. First, we demonstrate that the optimal weighting factor for recognizing the native structure differs from target sequence to target sequence. Then, we present three novel threading methods which circumvent training dataset-based weighting factor optimization. The basic idea of the three methods is to employ different weighting factor values and finally select a template structure for a target sequence by examining characteristics of the distribution of scores computed by using the different weighting factor values. Interestingly, the success rate of our approaches is comparable to the conventional threading method where the weighting factor is optimized based on a training dataset. Moreover, when the size of the training set available for the conventional threading method is small, our approach often performs better. In addition, we predict a target-specific weighting factor optimal for a target sequence by an artificial neural network from features of the target sequence. Finally, we show that our novel methods can be used to assess the confidence of prediction of a conventional threading with an optimized constant weighting factor by considering consensus prediction between them. Implication to the underlined energy landscape of protein folding is discussed.

Proteins 2008; 73:581–596.

Key words: threading; protein structrure prediction; scoring function; weight optimization; protein folding.

INTRODUCTION

Threading or fold recognition is a protein tertiary structure prediction method which uses a known protein tertiary structure as a template of the modeling.[1–3] Unlike conventional homology modeling methods which use a tertiary structure of an apparent homologous protein to a target protein as the template, a threading method aims to recognize a template structure which have a similar fold but may not have a significant sequence similarity to the target sequence. There are entire groups of proteins which are not necessarily evolutionarily related but have similar folds, which may be resulted by constraints imposed by physics and chemistry of the polypeptide chains.[4,5] Threading methods have been improving over the last decade, as evidenced in a world-wide protein structure competition, CASP (Critical Assessment of Critical Assessment of Techniques for Protein Structure Prediction).[6,7]

Strategies to recognize a distantly related template structure for a target protein in threading algorithms can be roughly categorized into two types: methods in the first type make use of extensive sequence information typically in the form of profiles[8,9] or Hidden Markov Models,[10–12] while the other type employs structure information in order to compensate insignificant sequence similarity between a target sequence and a template structure.[1,2,13,14] In the latter type, a scoring function for assessing the compatibility between a target sequence to a template structure combines a sequence similarity score with structure related scores such as a secondary structure matching score,[15] a residue accessible surface compatibility score,[13,16] and a residue–residue contact potential score.[17,18] These scoring terms of different properties are combined linearly with associated weighting factors:

$$E_{\text{total}} = \sum_i w_i E_i, \qquad (1)$$

Grant sponsor: National Institutes of Health (National Institute of General Medical Sciences); Grant numbers: R01GM075004, U24 GM077905; Grant sponsor: National Science Foundation; Grant number: DMS 0604776; Grant sponsor: Purdue Research Foundation
*Correspondence to: Daisuke Kihara, Department of Biological Sciences, College of Science, Purdue University, West Lafayette, IN 47907. E-mail: dkihara@purdue.edu.
Received 9 November 2007; Revised 24 February 2008; Accepted 20 March 2008
Published online 12 May 2008 in Wiley InterScience (www.interscience.wiley.com).
DOI: 10.1002/prot.22082

where E_{total} is the energy or the score given to a template structure aligned with a target sequence, E_i is each individual energy term, and w_i is the weighting factor associated with E_i. Commonly, the weighting factors are adjusted so that the prediction success rate is maximized using a training dataset of target sequences and template structures.

Determining appropriate weighting factors is a key for a successful threading algorithm, but it is not a trivial task. Weight optimization is involved not only in threading but also in many other bioinformatics prediction algorithms, such as *ab initio* protein structure prediction,[19,20,21] protein–protein docking,[22,23] and protein–ligand docking[24,25] algorithms. These scoring functions often combine empirical scoring terms of very different properties which do not have the same unit (e.g. a sequence similarity term and a knowledge-based amino acid contact potential), thus weighting factors need to be determined in a somewhat arbitrary way. Typically weighting factors are tuned so that a predefined error function of the prediction is reduced using an existing minimization algorithm, such as Monte Carlo simulated annealing[26] and downhill simplex algorithm,[23,27] or sometimes weighting factors are tuned manually by trial and error. In practice, the weight optimization is a very tedious and computationally expensive step because essentially each set of weighting factors are tested against the entire large benchmark dataset.

To find the optimal weighting factors rigorously and automatically, several methods have been proposed. Linear programming was applied to optimize weights so that correct templates for targets are distinguished from a negative set.[28] Lengauer and coworkers developed algorithms for optimizing parameters iteratively, which were applied to threading[29] and protein–ligand docking prediction.[30] Rosen *et al.* developed an algorithm which applies a sequential quadratic programming for searching global optimal parameters for a scoring function with a continuous degree of freedom.[31] There are also several works which optimize values in a contact potential.[32–35] Note that all the works mentioned earlier are optimizing parameters so that the resulting scoring function maximizes successes on a given benchmark dataset. A potential problem of training weighting factors on a benchmark dataset is that the weighting factors could be biased to the training set and may not be suitable for totally unseen target sequences. Of course in the ideal scenario, perfect weighting factors are obtained if a benchmark dataset well represents all variations of proteins and if the linear combination of chosen different scoring terms captures the "true" potential function at least approximately, although this is not the case in reality. More fundamentally it could be even argued that the justification of a linear combination of multiple scoring terms is not solid as the scoring terms are mostly nonphysical and may not be fully independent.

Here we explore different approaches to handle weighting factors for a scoring function of threading. Throughout this study we use a model system of gapless threading using a scoring function with two terms combined, a main chain angle potential and a residue contact potential, by a weighting factor. First, we demonstrate that the optimal weighting factor for recognizing the native structure differs from sequence to sequence, which questions the use of a constant weighting factor for all target sequences as the conventional threading methods do. Then, we present three novel threading methods which avoid optimization of the weighting factor based on a training dataset. The basic idea of the three methods is to employ different weighting factor values and finally select a template structure for a target sequence by examining characteristics of the distribution of scores obtained by the weighting factor values used. Interestingly, the success rate of our approaches is comparable, if not better, to the conventional threading method which uses the weighting factor optimized based on a training data set. Actually when the size of the training set is small on which the conventional threading method rely for optimizing the weighting factor, our approach often shows a better performance. In addition, we predict a target-specific weighting factor optimal to a target sequence by an artificial neural network (ANN) from features of the target sequence. Finally, we show that our methods can be used to assess the confidence of prediction of a conventional threading with an optimized constant weighting factor by considering consensus prediction between them. The methods introduced here bring novel ideas for weight optimization in protein structure prediction algorithms, which will have a large impact on different areas in bioinformatics and beyond.

MATERIALS AND METHODS

Data sets

3704 representative protein structures are selected for a benchmark dataset using the PDB-REPRDB server[36] with the default setting. First, membrane proteins, complex structures, and the structures solved by NMR are eliminated. Then, proteins are pruned by a sequence identity threshold of 30% and a root-mean square deviation (RMSD) of 10 Å. This dataset is used for gapless threading benchmark tests with a scoring function which combines a main-chain angle propensity potential and a residue contact potential. The representative proteins are classified by whether the structure is successfully recognized by its own sequence with the angle potential alone or the contact potential alone: Among the 3704 proteins, 2687 of them are recognized both with the angle potential alone and the contact potential alone. This subset of proteins is named db-A (Table I). The db-B set contains 398 proteins whose native structure is recognized with the angle potential alone but not with the contact potential alone. Oppositely, db-C contains 203 proteins whose native structure is recognized with the contact potential alone but not

Table I
Protein Benchmark Dataset

	Successful by the contact potential	Unsuccessful by the contact potential	Total
Successful by the angle potential	2687 (db-A)	203 (db-B)	2890
Unsuccessful by the angle potential	398 (db-C)	416 (db-D)	814
Total	3085	719	3704

Total of 3704 proteins are classified into four categories (db-A to db-D) by whether it is successfully recognized either by the angle potential alone (E_{ang3}) or by the contact potential alone.

with the angle potential alone. Finally, db-D contains 416 proteins whose native structure is not recognized neither with the contact potential alone nor the angle potential alone. We denote the whole database as db-ABCD. The proteins in db-A are omitted in some of the experiments later because a linear combination of the two potentials with any weighting factor value can successfully recognize the native structure. The dataset without proteins in db-A is denoted as db-BCD.

Main-chain angle potential

First we describe two scoring terms used in the scoring function of threading, namely, a knowledge-based statistical main-chain angle potential and a knowledge-based statistical residue-contact potential.

We consider a Cα model of a protein structure where adjacent Cα atoms are connected by a virtual bond (Fig. 1). The hinge angle of the *i*-th Cα atom, θ_i, is defined as an angle formed by three consecutive Cα atoms, $C\alpha_{i-1}$-$C\alpha_i$-$C\alpha_{i+1}$ ($0^\circ \leq \theta_i < 180^\circ$). Because the hinge angle is restricted by positions of covalent bonds of a carbon atom, the angle values have a skewed distribution with a mean value of 106.2°. The torsion angle of Cα, τ_I, is the dihedral angle defined by four consecutive Cα atoms, $C\alpha_{i-2}-C\alpha_{i-1}-C\alpha_i-C\alpha_{i+1}$ ($-180^\circ \leq \tau_i < 180^\circ$). Figure 2 shows an example of distribution of the hinge and the torsion angle of alanine. The dense region with the torsion angles between 0° to 90° and the hinge angle at around 130° represents the propensity of Alanine to be included in a β strand, while the region around $\tau_i = \tau_{i+1} = -120^\circ$ corresponds to α helices. The region with $\tau_i = -120^\circ$ and τ_{i+1} between 0° to 90° represents the C-terminal end of α helices [Fig. 2(B)].

We define the residue-dependent main-chain angle potential, E_{ang3}, which describes the propensity of an amino acid type α having a certain range of hinge angle and two torsion angles:

$$E_{\text{ang3}}(\alpha, \tau_1, \theta, \tau_2) = -\ln \frac{p(\alpha, \tau_1, \theta, \tau_2)}{p(\tau_1, \theta, \tau_2)}, \tag{2}$$

where $p(\alpha,\tau_1,\theta,\tau_2)$ is the probability that the amino acid α has a hinge angle of θ, a preceding torsion angle of τ_1, and a succeeding torsion angle of τ_2. The denominator is the probability that any amino acid takes the combination of the angles of θ, τ_1, and τ_2, simultaneously. The entire range of the hinge angle and the torsion angle is divided by a bin size of 5° and 10°, respectively, thus the whole angle space of θ, τ_1, and τ_2 is divided into $(180/5) \times (360/10) \times (360/10) = 46{,}656$ cells. The angle values are sampled from proteins of the 9/10 of db-ABCD, and the gapless threading results of the angle potential alone shown in Table II is tested on the rest of 1/10 of the proteins. Because some of the cells have too few data points, Kernel Density Estimation (KDE)[37] is used to estimate the true distribution of the angles from the sampled data. KDE essentially smoothes data distribution of the sampled data points. The Gaussian kernel function with the smoothing parameter or the bandwidths of (15, 8, 15) are used for (τ_1, θ, τ_2).

In addition to E_{ang3}, another angle potential, E_{ang2}, which concerns two angles, the hinge angle, θ, and only the preceding torsion angle, τ_1, is derived:

$$E_{\text{ang2}}(\alpha, \tau_1, \theta) = -\ln \frac{p(\alpha, \tau_1, \theta)}{p(\tau_1, \theta)} \tag{3}$$

KDE is used with the parameters of (5, 3) for (τ_1, θ). As will be shown in Table II, the performance of E_{ang2} in the gapless threading was worse than E_{ang3}. Therefore, we combine E_{ang3} to the contact potential for the threading scoring function used throughout this study.

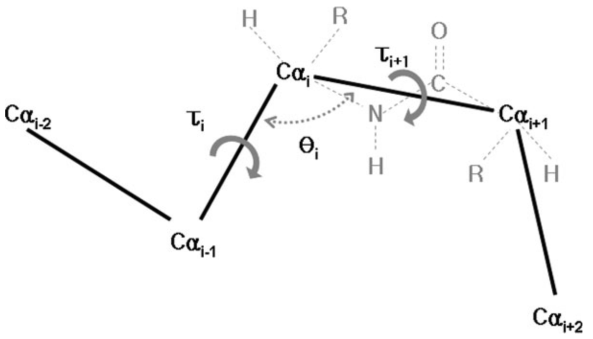


Figure 1
The definition of the hinge angle, θ_i, and the torsion angle, τ_I, of the Cα model of a protein.

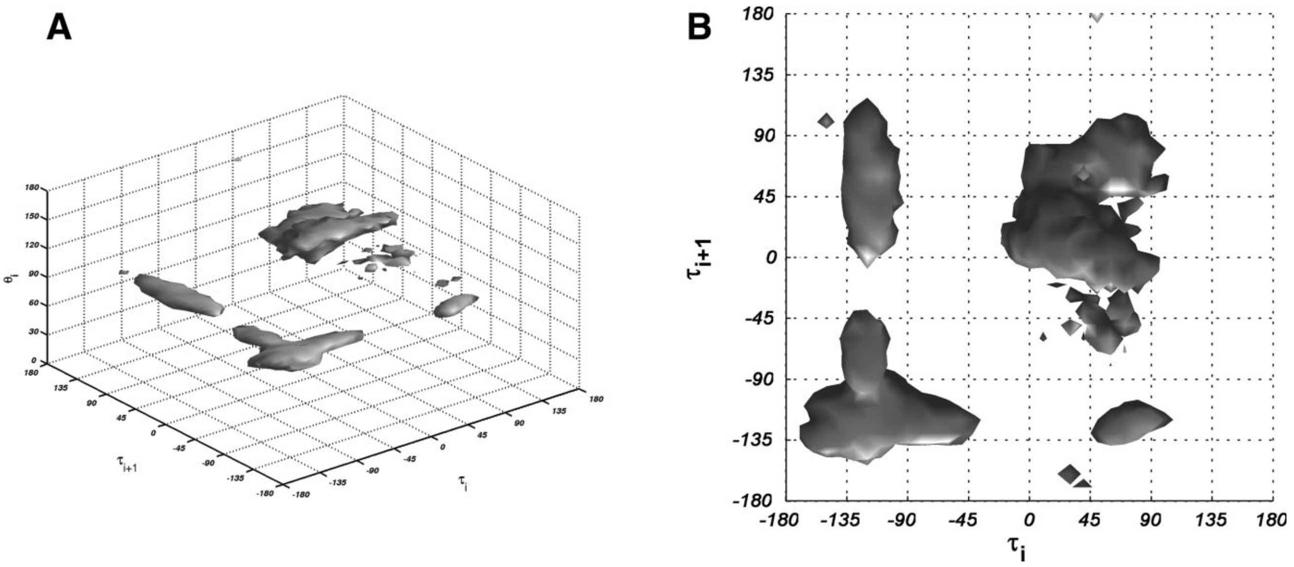


Figure 2

The distribution of the hinge and the torsion angles of alanine, which are sampled from the db-ABCD data set. The frequency of the data points in 36 × 36 × 36 cells are smoothed using KDE with the smoothing parameters of (15, 8, 15) for (τ_i, θ_i, τ_{I+1}). The iso-surface of the frequency of 2.0 × 10^{-4} is shown. Note that the average frequency of a cell is 1/(36 × 36 × 36) = 2.14 × 10^{-5}. **B** is the same data as **A** viewed from the top (the θ axis).

Residue contact potential

Knowledge-based pairwise contact potentials have been shown effective in many areas of protein structure research including threading,[17,18,38–40] *ab initio* structure prediction,[19,41,42] and quality assessment of predicted protein structure models.[43–46] As the second term in the scoring function of the threading, we use a knowledge-based residue–residue contact potential computed using a quasi-chemical approximation[17] for the reference state and a contact definition of 4.5 Å between any side-chain heavy atoms. Thus, the contact potential for a given pair of an amino acid pair of α and β is given by:

$$E_{\mathrm{contact}}(\alpha,\beta) = -\ln\frac{p(\alpha,\beta)}{p(\alpha)p(\beta)}, \qquad (4)$$

where $p(\alpha, \beta)$ is the observed frequency of contacts between amino acid α and β, $p(\alpha)$ and $p(\beta)$ are the frequency of single amino acid, α and β, respectively, in a protein structure data set. Both the angle potential and the contact potential are available at our website: http://dragon.bio.purdue.edu/nonopt_suppl/.

Linear combination of the angle potential and the contact potential

E_{ang3} and E_{contact} are linearly combined to form the scoring function in order to capture local structure propensity of a sequence and distant residue–contact propensity[47]:

$$E_{\mathrm{total}} = E_{\mathrm{ang3}} + wE_{\mathrm{contact}} \qquad (5)$$

Here *w* is the weighting factor which balances the contribution of the contact potential and the angle potential.

Gapless threading procedure

We employ gapless threading[48–52] to evaluate a scoring function, E_{total}, which uses a weighting factor, *w*. Proteins in a protein library are virtually connected to form a long

Table II

Summary of the Success Rate Using Different Methods

Methods	Top 1[a]	Top 5
Angel potential (E_{ang2}) alone	69.0 (%)	75.1 (%)
Angle potential (E_{ang3}) alone	78.0	83.5
Contact potential (E_{contact}) alone	83.3	88.0
Optimal constant weighting factor[b]	91.7	94.8
Optimal target-specific weighting factor[c]	93.6	95.8
Top rank frequency method	91.4	93.9
Smallest *Z*-score method	90.6	94.5
Largest *Z*-score gap method	89.9	93.4
Neural network[d]	91.3 (0.07)	94.6 (0.04)

[a] Top1, the percentage of the proteins whose native structure is recognized in the top one position; Top 5, a prediction is counted as accurate when the native structure is recognized in the top five positions.

[b] Training of the weighting factor method here is done on the whole db-ABCD dataset, and so is the threading test.

[c] For each of the target sequence, the preferable weighting factor which gives the correct prediction is used (if any).

[d] The weighting factor preferable for a target sequence is predicted by neural network. The number in the parenthesis is the standard deviation of the success rate for 200 different pairs of training set (3350 proteins) and testing set (354 proteins).

polypeptide chain on which a target sequence is aligned through by shifting by one residue at a time. When a target sequence is aligned on multiple templates, interaction between the template structures is not considered by the contact potential. For a target sequence, the score E_{total} is computed at every position along the virtual template polypeptide, and a template which has not less than 50% overlap with the target is considered to be recognized by the target. When the whole template dataset, db-ABCD, is used, the number of alternative positions is 778,687.

Threading methods without optimizing weighting factor by training

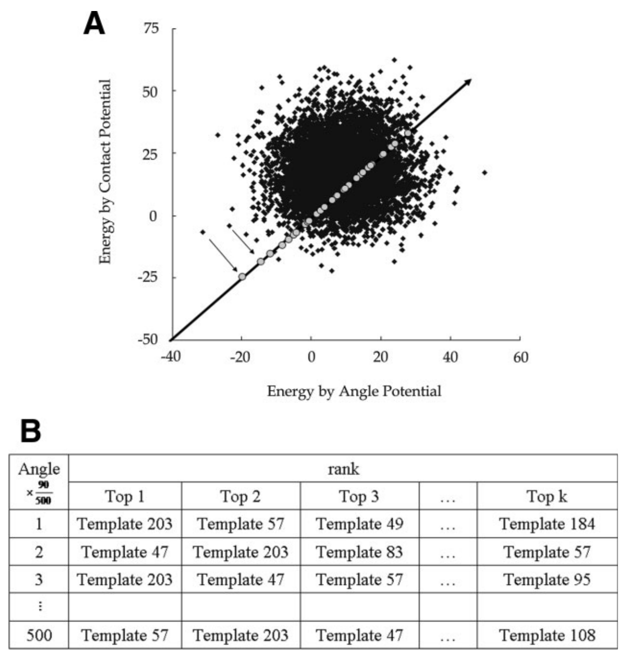
We introduce three novel threading approaches without using an optimized weighting factor for a training dataset. In addition, another approach that uses predicted target-specific optimal weighting factor for each particular target sequence is described. For convenience, in what follows we denote the conventional threading approach which uses a weighting factor optimized based on a training dataset as threading$_{\mathrm{const\_weight}}$.

Top rank frequency method

The score of the two terms, E_{ang3} and E_{contact}, [Eq. (5)] for a match of a target sequence and a template structure can be represented in a two dimensional plot [Fig. 3(A)]. Computing E_{total} is considered as the projection of the data point onto the line with a slope of arctan(w), which ranges from 0° to 90°. Thus, changing the weight w is interpreted as changing the slope of the line on which data points of templates are projected. The template structure with the best (smallest) score is the one which is projected to the closest position to the intersection of the x axis and the y axis. When the slope is 0°, only E_{ang3} contributes to E_{total}, and only E_{contact} contributes to E_{total} when the slope is 90°.

The basic idea of the Top Rank Frequency Method is to try all possible weighting factor values for a given target sequence and consider how many times each template structure enters in top ranks [Fig. 3(B)]. More concretely, first, for a target sequence, E_{ang3} and E_{contact} of each template are calculated. Because the gapless threading is performed, E_{ang3} and E_{contact} can be computed separately without considering the weighting factor. Then, the slope angle of the projection line is altered from 0° to 90° in 500 steps, and for each slope angle value, E_{total} of each template is calculated and ranked, resulting in 500 ranking lists of templates [i.e. 500 rows in the table in Fig. 3(B)].

Two flavors of the top ranking frequency method are designed. In the first method, comparison of templates starts from the top 1 level to the top k level until one of them has more counts at a specific level than the other structures. This method is termed the Olympic Ranking method, because the procedure resembles the ranking of countries in an Olympic Games based on the number of gold, silver, and bronze medals acquired. The second method is termed the top 10 ranking method. Here each template is scored one point for each time it is ranked within top 10, and the one which scores the most is the final choice of the template for a target sequence [Fig. 3(B)].



Angle $\times\frac{90}{500}$	rank				
	Top 1	Top 2	Top 3	…	Top k
1	Template 203	Template 57	Template 49	…	Template 184
2	Template 47	Template 203	Template 83	…	Template 57
3	Template 203	Template 47	Template 57	…	Template 95
⋮					
500	Template 57	Template 203	Template 47	…	Template 108

Figure 3

The gapless threading procedures used in this study. (**A**) A schematic representation of gapless threading with a scoring function with two terms, an angle potential (E_{ang3}) and a residue contact potential (E_{contact}). (**B**) Illustration of the Top Rank Frequency Method, the Smallest Z-score Method, and the Largest Z-score Gap Method. For each time the weighting factor w is changed, the ranking of the templates ordered by E_{total} changes. In the Top Rank Frequency Method, the template which is ranked at Top 1 for the largest number of times is selected. If there is a tie by two templates, then the number of times the two templates are ranked at Top 2 is compared (the Olympic ranking method). In the top 10 ranking method of the Top Rank Frequency Method, the template which is ranked within Top 10 the most of the times is selected. In the Smallest Z-score Method, among the templates listed in the column of "Top 1," the one which has the smallest Z-score is selected. In the Largest Z-score Gap Method, among the templates listed in the column of Top 1, the one which has the largest Z-score gap between the template at Top 1 and Top 2 is selected.

Intuitively the Top Rank Frequency Method works successfully when a target sequence is compatible to its native structure far better than to the other template structures so that a small fluctuation of the weighting factor of the scoring function does not affect much to the recognition of the native structure.

Smallest Z-score method

Similar to the Top Rank Frequency Method, the Smallest Z-score Method starts from the 500 ranking lists of templates for a target sequence using 500 different weighting factors of the scoring function [Fig. 3(B)]. For each of the 500 ranking lists, the Z-score of the all template structures is computed. The Z-score of a template structure j, using a weighting factor i, Z_{ij}, is defined as follows:

$$Z_{ij} = \frac{E_{ij} - \overline{E}_i}{SD(E_i)}, \tag{6}$$

where E_{ij} is the total score of the template j computed using a weighting factor i. $\bar{E}_i$ and $\mathrm{SD}(E_i)$ are the average and the standard deviation of the total score of all the templates in a library computed with a weighting factor i, respectively. The Z-score of the top hit has a smallest negative value, because the smaller score is better in our scoring function [Eqs. (3) and (4)]. Since there are 500 ranking lists, a template has 500 different Z-scores at maximum. Among them, the smallest (the most significant) Z-score is assigned to the template, and templates are reranked based on the smallest Z-score assigned. Finally, the template with the smallest Z-score is selected.

The rationale of the Smallest Z-score Method is that the native structure is expected to have the significant Z-score if the proper weighting factor is used in the scoring function. Therefore, in this method, a weighting factor is sought which gives the smallest Z-score to one of the templates.

Largest Z-score gap method

In this method, to each of the top ranking template in the 500 ranking lists, the difference between the Z-score of the top ranked template and the second ranked template is assigned. Then, the template which has the largest Z-score gap assigned is finally selected. The motivation of the Largest Z-score Gap Method comes from the discussion in theoretical studies of protein folding that the protein sequence and structure is evolved in a way that the energy of the native structure has the large gap between the next stable structure in order for the conformation of the native structure to be quickly found in the vast folding landscape and remains stable.[53]

Execution of the above three methods

Given a target sequence, E_{ang} and E_{contact} are separately precomputed for each template. Then these two scores of a template are linearly combined by 500 different weighting factors, yielding 500 rank lists of templates [Fig. 3(B)]. Then, the final template is selected following each strategy. Therefore, actual number of threading computation for a target is 2 × N, where N is the number of templates and 2 is the number of scoring terms used here, instead of $N \times M$, where M is the number of alternative values of the weighing factor (in this study, $M = 500$).

Predicted weighting factor

At last, we introduce a different type of approach, which predicts a preferable weighting factor specific to a target sequence by considering features of the sequence using an ANN. A 3-layer feed-forward neural network is used with 21 nodes for the input layer, 11 nodes for the middle layer, and 1 node for the output layer. Input parameters used are the amino acid composition (using 20 input nodes) and the effective length, L_{eff}[54] (for 1 input node). We used the amino acid composition because it is relevant to the structural class of a protein, for example α class and α/β class.[55–57] L_{eff} is defined as follows:

$$\begin{aligned} L_{\mathrm{eff}} = {} & (\text{number of } \alpha \text{ helices and } \beta \text{ strands}) \\ & + (\text{number of residues in coil regions}) \end{aligned} \tag{7}$$

Here a coil is simply defined as a region which is included neither in an α helix nor in a β strand. L_{eff} of a protein is shown to have a significant correlation to the folding rate.[54] We use the effective length with an expectation that a protein with a small L_{eff} value (thus with a relatively large secondary structure content) might prefer a larger weight to the angle potential and a protein with a large L_{eff} might need a larger weight for the contact potential. The secondary structure used to compute L_{eff} of a target protein is predicted by SABLE,[58] that is, L_{eff} is predicted.

MATLAB R12 is used to construct the ANN. The ANN is trained by the back-propagation algorithm.[59] The transfer function from input layer to middle layer is a *tansig* function and the transfer function from middle layer to output node is a *logsig* function. Levenberg-Marquardt method[29] is used to train the ANN and the Bayesian regularization method[30] is used to stop training before parameters are over-fitted.

RESULTS

Angle potential and contact potential

First we examined how many protein sequences in db-ABCD can recognize its native structure using a single potential, either with the angle potential alone or the contact potential alone (Table II). The accuracy by E_{ang3} is 78.0% in the top position and 83.5% in the top five scoring positions. E_{ang2} performs slightly worse, 69.0% in the top position and 75.1% in the top five positions. In either case, the performance of the angle potential is quite good, which qualitatively agrees with some early studies which report local structure propensity of amino acids has a dominant effect in determining protein topology.[60,61] However, the flip-side of the results is that the rest of more than 20% of the proteins tested were not recognized by the angle potential only. Local structure propensity of amino acids is important but not sufficient in determining protein topology, as clearly shown by our previous work.[62] Indeed, in this test, the contact potential performs better than the angle potential, recognizing 83.3% in the top position and 88.0% in the top five positions.

Threading with the optimal constant weighting factor

Next we investigate how well the conventional threading method which uses a constant weighting factor optimized on a training dataset performs. To simulate the best possible scenario that we have the perfect training dataset for determining the weighting factor, that is the case when the training set is identical to the test set, we used the whole dataset, db-ABCD, to compute the optimal constant weighting factor. Concretely, we compute the success rate by using all 500 different weighting factor values and choose the one which gives the highest success rate. The success rate is 91.7% in the top position and 94.8% in the top five positions, both of which are higher than those by the contact potential or the angle potential alone.

Ultimate performance by the target-specific optimal weighting factor

The last row of the upper half of Table II shows the results of threading using the optimal weighting factors specific for each target sequence. For each target protein, all possible 500 values of the weighting factor are tested and the one which can successfully recognize its native structure is employed. Thus the weighting factor used can differ for each target sequence. The success rate is 93.6% in the top position and 95.8% in the top five positions. Of course in a real prediction, it may not be possible to find the target sequence-specific optimal weighting factor as we demonstrate here. The purpose of showing this ultimate performance is to consider the upper-bound of the success rate by the current scoring function. The important thing to note is that the success rate of the threading with the optimal target-specific weighting factor is higher than that of the threading with the optimal constant weighting factor. This comparison indicates that there are protein sequences which need a specific weighting factor for recognizing its native structure that does not agree with the majority of protein sequences.

Figure 4 shows the distribution of the optimal target-specific weighting factor of the 1017 proteins in db-BCD. In the case that several optimal weighting factor values exist for a target protein that can recognize its native structure, all of them are counted. Proteins in db-A is omitted because any value of the weighting factor works to recognize the native structure. This distribution clearly shows that each protein has its own optimal weighting factors, so that a constant weighting factor can not be optimal for all target protein sequences.

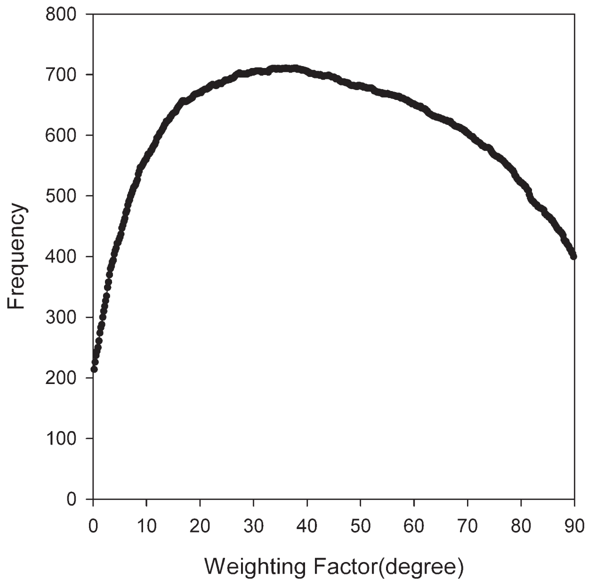


Figure 4

The distribution of preferable weighting factors for target sequences in db-BCD that make the scoring function, E_{total}, capable of identifying the native structure. The x-axis is the slope of the line in Figure 2, that is arctan (w). If several weighting factors are preferable by a target sequence, all of them are counted.

The difference of the success rate between the threading with the optimal constant weighting factor and the threading with the optimal target-specific weighting factor is our motivation to explore alternative approach to handle weighting factors in a scoring function of threading. A direct approach to realize the target-specific weighting factor is to "predict" the optimal weighting factor for a given target sequence, which is attempted by an ANN in this study.

Threading methods without using the optimized constant weighting factor

Now we analyze the results of the four methods we introduce in this study, namely, the Top Rank Frequency Method, the Smallest *Z*-score Method, the Largest *Z*-score Gap Method, and the threading with the target-specific weighting factor predicted by ANN (the bottom half of Table II). The first three methods show a similar performance, although the Top Rank Frequency Method is slightly better than the other two when templates recognized at the top position are considered. Figure 5 further compares the three methods in the top 1, top 5, 10, and 20 positions. The Top Rank Frequency Method recognizes a template correctly in 91.4% of the cases at the top 1 position, in 94.6% of the cases within top 5, and in 95.2% of the cases within the top 10 positions. For

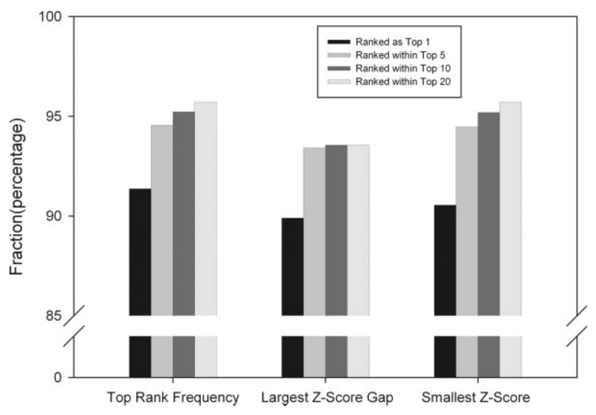


Figure 5

The performance of the Top Rank Frequency Method, the Smallest *Z*-score Method, and the Largest *Z*-score Gap Method. The percentage of target sequences in db-BCD which recognized its native structure within the top 1, 5, 10, and 20 ranks are shown.

the Top Rank Frequency Method to work, a target protein should have a wide range of preferable weighting factors so that the native structure obtains sufficient votes to stand out. Figure 6 shows that actually that is the case; the target sequences whose native structure is successfully recognized by the Top Rank Frequency Method have clearly a wider range of preferable weighting factors [Fig. 6(A)] compared with the ones which are successful by the threading with the optimal constant weighting factor but not by the Top Rank Frequency Method [Fig. 6(B)].

It is surprising that these three methods performed almost as good as the threading with the optimal constant weighting factor (Table II). Here be reminded that the threading with the optimal constant weighting factor in Table II is using the weighting factor which yields the highest success rate. In contrast, our three methods are obtained without any prior knowledge about the optimal weighting factor from training. Fairer head-to-head comparison against the threading with a trained constant weighting factor will be discussed in the next section.

At the last row of Table II, we show the success rate of the threading using predicted a weighting factor by ANN. The success rate shown is the average of 200 different pairs of training and testing sets randomly chosen from db-ABCD. The small standard deviation in the parenthesis indicates that the results are stable to the training and testing datasets used. The average success rate is 91.3% in the top one position and 94.6% in the top 5 positions, which are quite close to those of the threading with the optimal constant weighting factor. These encouraging results show the proof of concept that a correlation between a sequence and the preferable weighting factor exists, which can be learned by ANN. However, there is still a room between the success rate achieved by the ANN approach and the results by the threading with the optimal target-specific weighting factor (93.6% in Top 1). Here the input parameters used for ANN approach are a simple combination of the amino acid composition and the effective length of a target protein. Further investigation of appropriate input parameters will lead to an increase of the success

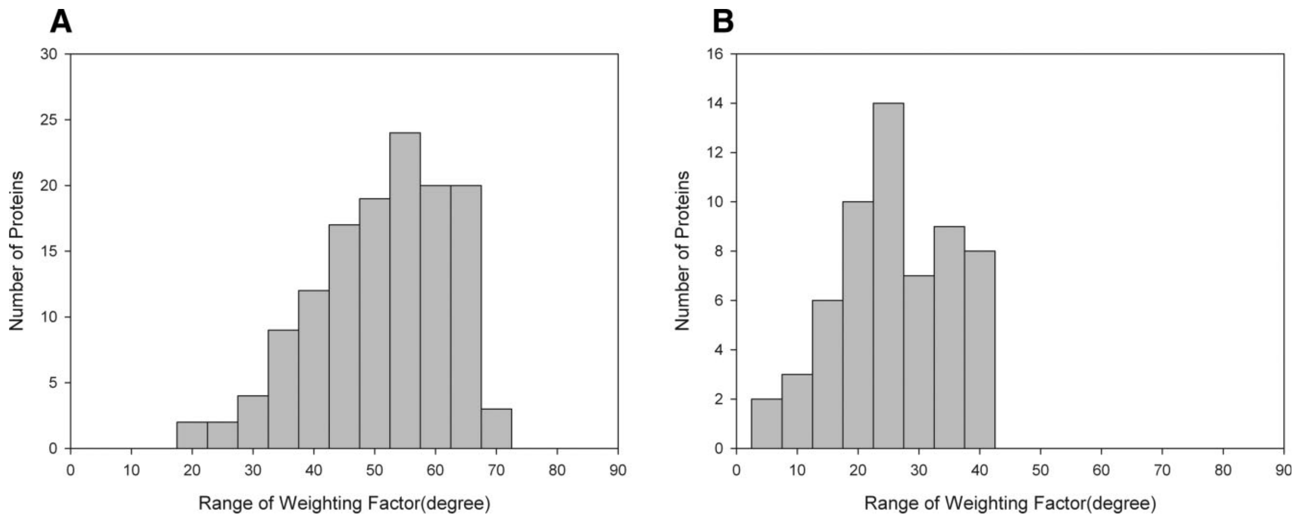


Figure 6

The range of preferable weighting factors [arctan(*w*)] of target proteins whose native structure is recognized (**A**) by the Top Rank Frequency Method but not by the threading with the optimal constant weighting factor; and (**B**) by the threading with the optimal constant weighting factor but not by the Top Rank Frequency Method. The db-BCD set is used.

rate. Using a different advanced machine learning algorithm may also help.

Head-to-head comparison between top rank frequency method and the conventional threading

In the previous section, we have demonstrated that the four newly introduced methods have a comparable success rate to the threading with the optimal constant weighting factor. In this section, we pick up the Top Rank Frequency Method, which shows the best performance in the top one position among the four methods (see Fig. 5), and further compare it with the conventional threading with a constant weighting factor (threading$_{\mathrm{const\_weight}}$). Here we separate a training dataset from a testing set, and the constant weighting factor for the conventional threading is optimized based on the training dataset. We have conducted two types of comparisons. In the first test, named the two-partition test, the size of the training and the testing datasets is altered to examine how the training data size affects to the success rate. In the second test, named the 100–100 test, an equal number of proteins are used for the training and testing data set.

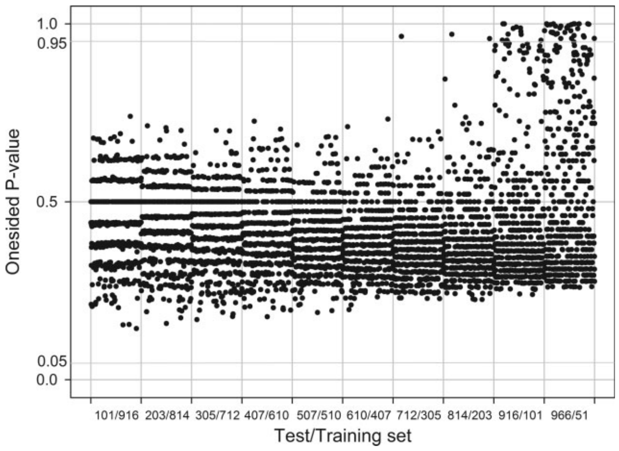


Figure 7

Two-Part Partition Tests performed to compare Top Rank Frequency Method and threading$_{\mathrm{const\_weight}}$. The *P*-value, P^{I}_{jk} is plotted for each partition (*j*) and test (*k*).

Two-partition test

The two-sample proportion test is performed to compare the success rates of the Top Rank Frequency Method (p_1) and for threading$_{\mathrm{const\_weight}}$ (p_2).[63] The total of 1017 proteins in the db-BCD set are partitioned into two groups, one for the training set and the other for the testing set. The number of proteins in the training and the testing sets is altered in 10 different ratios as follows: Let n_k be the size of the testing set for the k-th testing/training ratio ($k = 1,\ldots,10$). Because the total number of proteins is 1017, the size of the training dataset is $1017 - n_k$.

$n_1 = 101$ (916), $n_2 = 203$ (814), $n_3 = 305$ (712), $n_4 = 407$ (610), $n_5 = 507$ (510),

$n_6 = 610$ (407), $n_7 = 712$ (305), $n_8 = 814$ (203), $n_9 = 916$ (101), $n_{10} = 966$ (51).

In the parentheses, the number of proteins in the training set is shown. We randomly choose n_k number of proteins from db-BCD for a given testing/training ratio k ($k = 1,2,\ldots,10$). Then, the number of successes among n_k test proteins made by the two methods, Top Rank Frequency Method and threading$_{\mathrm{const\_weight}}$ is counted. This procedure is repeated 500 times for each of the 10 testing/training ratios.

Let X_{ijk} ($i = 1,2$; $j = 1,2,\ldots,500$; $k = 1,2,\ldots,10$) be the number of successes among n_k number of predictions made in the j-th data set for the k-th testing/training ratio. Here, $i = 1$ corresponds to the Top Rank Frequency Method, and $i = 2$ corresponds to threading$_{\mathrm{const\_weight}}$. Then the test statistic for the two-sample proportion problem is defined as

$$T_{jk} = \frac{\hat{p}_{1jk} - \hat{p}_{2jk}}{\sqrt{2\hat{p}_{.jk}(1 - \hat{p}_{.jk})/n_k}}, \qquad (8)$$

where $\hat{p}_{ijk} = X_{ijk}/n_k$, $i = 1,2$, and $\hat{p}_{.jk} = (X_{1jk} + X_{2jk})/(2n_k)$.

The hypotheses to be tested are two types:

$$\begin{aligned} &\mathrm{I})\, H_0 : p_1 = p_2, \quad H_A : p_1 < p_2 \\ &\mathrm{II})\, H_0 : p_1 = p_2, \quad H_A : p_1 > p_2 \end{aligned}$$

The corresponding *P*-values are $P^{I}_{jk} = P(Z > T_{jk})$ for the first hypotheses (I), and $P^{II}_{jk} = P(Z < T_{jk})$ for the second hypotheses (II), where Z denotes the standard normal random variable. Note that $P^{II}_{jk} = 1 - P^{I}_{jk}$. The *P*-values P^{I}_{jk} are plotted in Figure 7 with 5% and 95% horizontal reference lines. If a point lies below the 5% horizontal line, it means that the null hypothesis in hypotheses (I) will be rejected, and thus threading$_{\mathrm{const\_weight}}$ is significantly better than the Top Rank Frequency Method. If a point lies above 95% horizontal line, on the other hand, it means that the null hypothesis in hypotheses (II) will be rejected, and thus the Top Rank Frequency Method is significantly better than the conventional method in prediction. In Figure 7, we see that no point is below 5% line for any of the testing/training ratio, but some points exist (although few compared to the total number of comparisons, 500) are above 95% line for testing/training ratio of 712/305, 814/203, 916/101, and 966/51. Especially when testing/training ratio is 966/51, there are a substantial number (32 of 500) of *P*-values above 95% reference line. From the statistical analyses, we conclude that the

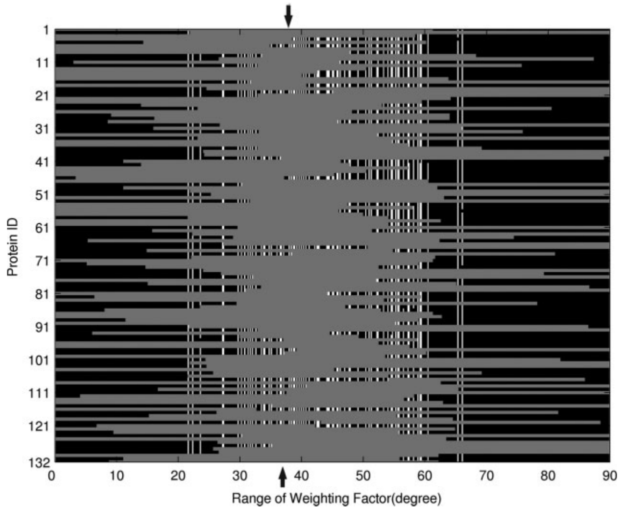


Figure 8

Preferable weighting factors of target proteins whose native structure is recognized by the Top Rank Frequency Method but not by threading$_{const\_weight}$. Preferable weighting factors of each of the 132 target proteins are shown as gray regions on the horizontal axis. White vertical lines represent the optimized weighting factors for threading$_{const\_weight}$ computed based on a training set. Since the value of the optimized weighting factor depends on the training dataset used, multiple weighting factors are shown. For reference, the arrows indicate the optimal constant weighting factors (Table II), which is the best weighting factor for the whole db-ABCD set.

Top Rank Frequency Method is at least as good as threading$_{const\_weight}$ in prediction and tends to perform better when the size of the training set available for threading$_{const\_weight}$ is small.

In Figure 8, we further examine concrete examples of preferable weighting factors of target proteins whose native structure is recognized by the Top Rank Frequency Method but not by threading$_{const\_weight}$. The tests are repeated 500 times with a training set of 101 proteins and a testing set of 916 proteins. There are 132 such proteins which are not recognized at least once among the 500 tests. All of these target proteins have a wide range of preferable weighting factors, but that range does not include the one of the optimized weighting factors indicated by white vertical lines in the Figure 8. Some of the target proteins shown here also are not recognized by the threading with the optimal weighting factors indicated by the arrows (Table II). These proteins do not belong to a specific fold type: 40 of them belong to α class, 21 are β class proteins, 12 proteins belong to α/β class, 24 proteins belong to α+β class, and the rest of them are categorized into the other classes in the SCOP database.[64] The Top Rank Frequency Method successfully captures native structures of those target proteins which do not follow the preference of the weighting factor of the majority of the target proteins by considering the range of the preferable weighting factors, but not its actual value.

100–100 test

The next experiment we present is a head-to-head comparison of the Top Rank Frequency Method with threading$_{const\_weight}$. From db-BCD, two sets of 100 proteins are randomly selected, one for a training set and the other for a testing set (testing/training ratio = 100/100). The weighting factor of threading$_{const\_weight}$ is optimized based on the training set and the performance of the two methods on the testing set is examined. We followed exactly the same procedure as we did in two-partition test for the given testing/training ratio: the number of successes among 100 predictions made by the two methods, Top Rank Frequency Method and threading$_{const\_weight}$ is counted. This procedure was repeated to generate 500 data sets. The two-sample proportion test was made for each of the 500 data sets and the corresponding p-values are plotted in Figure 9. The way to examine Figure 9 is the same as Figure 7. In Figure 9, no point lies below 5% line or above 95% line, which means that none of the two methods is significantly better than the other, as we may expect from Figure 7.

Consensus prediction between two methods

Next issue we investigate is consensus prediction by threading$_{const\_weight}$ and either one from the Top Rank Frequency Method, the Smallest *Z*-score Method, or the Largest *Z*-score Gap Method (Fig. 10). As is done in Figure 7, the db-BCD set is separated into two parts, one for testing set and the other one for the training set with 10 different ratios. Given a pair of testing and training sets, threading is performed on the testing set using threading$_{const\_weight}$ and also either one of the Top Rank

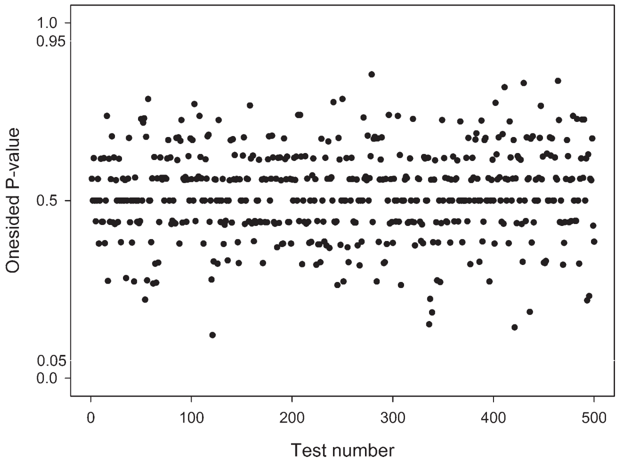


Figure 9

Prediction performance comparison of threading$_{const\_weight}$ and the Top Rank Frequency Method in the 100–100 test.

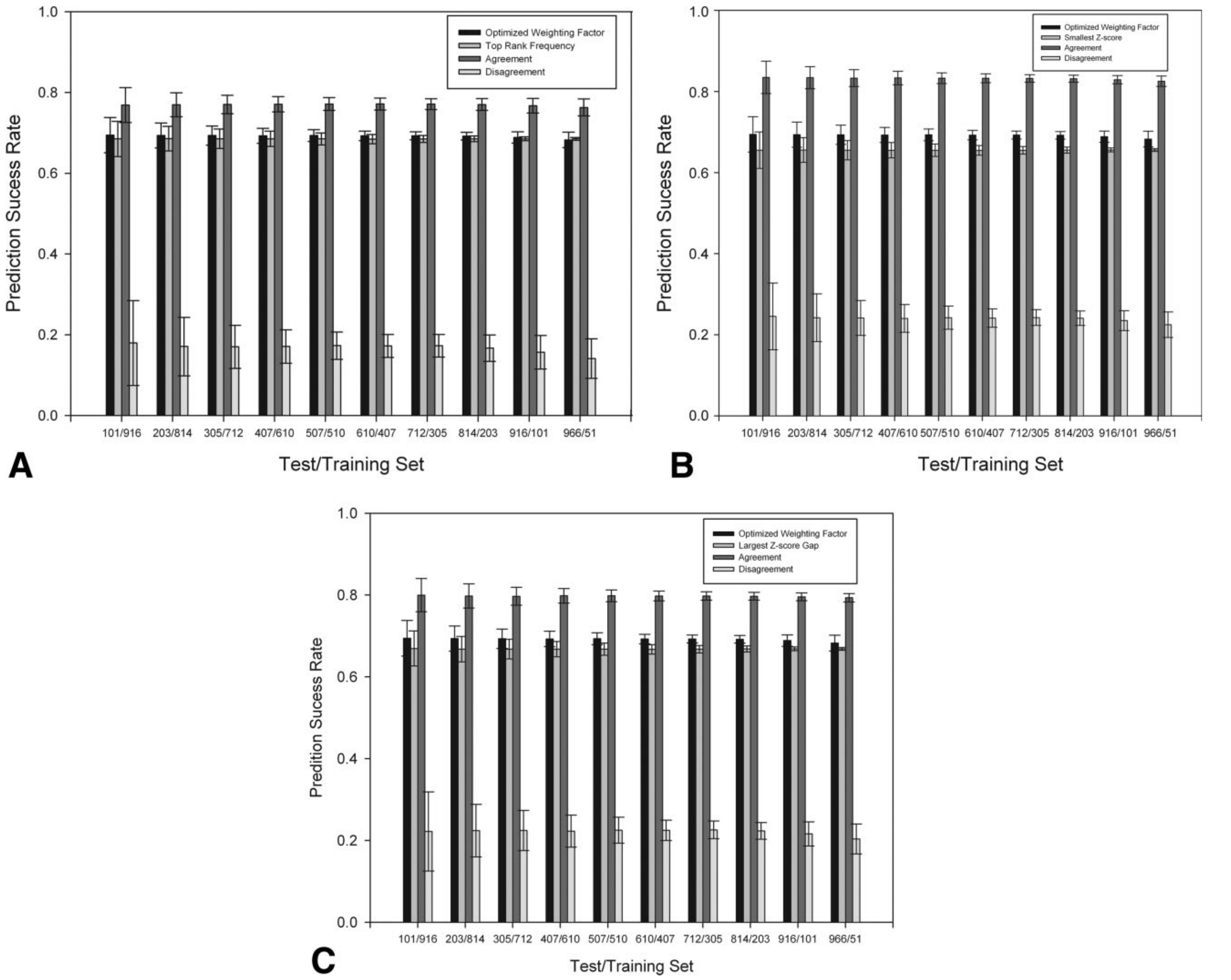


Figure 10

The success rate of the consensus prediction by threading$_{\mathrm{const\_weight}}$ and Top Rank Frequency Method, the Smallest *Z*-score Method, or the *Z*-score gap method. Bars from left to right: black bar, the success rate of threading$_{\mathrm{const\_weight}}$; gray bar, the success rate of the either one among the three proposed methods in this work; bar in dark gray, the success rate when both methods agree; bar in pale gray, the success rate when the two methods disagree. (**A**) Consensus between threading$_{\mathrm{const\_weight}}$ and the Top Rank Frequency Method; (**B**) consensus between threading$_{\mathrm{const\_weight}}$ and the Smallest *Z*-score Method; (**C**) consensus between threading$_{\mathrm{const\_weight}}$ and the Largest *Z*-score Gap Method.

Frequency Method, the Smallest *Z*-score Method, or the *Z*-score gap method. Then, the prediction success rate is computed for the cases when both methods agree and also disagree to select a template at the first position for a target sequence. For each testing/training set ratio, 500 different pairs of a testing and a training set are generated, and the average success rate of the 500 sets is computed.

Figure 10(A–C) shows the success rate of the cases of consensus and nonconsensus between threading$_{\mathrm{const\_weight}}$ and the Top Rank Frequency Method, the Smallest *Z*-score Method, and the Largest *Z*-score Gap Method. Remarkably, the success rate of the cases of consensus is significantly higher than that of prediction of individual method: In Figure 10(A), the average success rate (over the different testing/training set ratios shown on the X-axis) of the consensus prediction between threading$_{\mathrm{const\_weight}}$ and the Top Rank Frequency Method is 68.52%, which is 6.75 and 7.49 percentage points higher than the average success rate of threading$_{\mathrm{const\_weight}}$ alone and the Top Rank Frequency Method alone, respectively. The success rate of the consensus is highest with the Smallest *Z*-score Method (83.26%) [Fig. 10(B)], the second highest with the largest *Z*-score Gap Method (79.77%) [Fig. 10(C)] and the consensus with the Top Rank Frequency Method is the third [Fig. 10(A)]. So for example, if a template is selected by threading$_{\mathrm{const\_weight}}$ as having the smallest *Z*-score and if that *Z*-score is the smallest possible *Z*-score any template in the template database will be assigned by altering the weighting

Table III
The Success Rate of the Consensus Approach and the *Z*-Score Threshold Approach[a]

Size of testing/training dataset	Number of predictions by consensus approach	Number of predictions by *Z*-score threshold approach[b]	Overlap[c]	Success rate of consensus approach	Success rate of *Z*-score threshold approach
101/916	88.2 (3.1)	88.3 (3.0) *Z*: −4.34	81.1 (3.6)	76.89 (4.30)	76.21 (4.26)
203/814	177.2 (4.9)	177.2 (4.4) *Z*: −4.36	162.6 (5.4)	76.96 (2.95)	76.18 (3.04)
305/712	265.9 (6.3)	266.0 (5.2) *Z*: −4.36	243.9 (6.4)	77.03 (2.27)	76.29 (2.34)
407/610	354.1 (7.5)	354.1 (5.5) *Z*: −4.37	324.4 (6.9)	77.09 (1.91)	76.37 (1.83)
507/510	440.9 (9.3)	440.9 (5.5) *Z*: −4.36	403.9 (7.8)	77.13 (1.59)	76.40 (1.46)
610/407	529.7 (11.3)	529.7 (5.7) *Z*: −4.37	484.7 (8.1)	77.14 (1.49)	76.38 (1.19)
712/305	618.7 (13.5)	618.7 (5.9) *Z*: −4.37	566.5 (8.8)	77.12 (1.35)	76.34 (0.98)
814/203	708.6 (17.3)	708.6 (5.9) *Z*: −4.36	649.5 (10.8)	77.01 (1.46)	76.13 (0.89)
916/101	797.9 (22.1)	797.9 (5.9) *Z*: −4.34	732.1 (15.0)	76.72 (1.81)	75.66 (1.15)
966/51	841.2 (27.0)	841.2 (17.4) *Z*: −4.32	771.6 (21.7)	76.29 (2.10)	74.96 (1.62)

[a] The consensus approach is to only consider the predictions when threading$_{\text{const\_weight}}$ and the Top Rank Frequency Method agree. The *Z*-score threshold approach is to only consider the predictions by threading$_{\text{const\_weight}}$ when the *Z*-score is smaller (more significant) than the threshold value. The db-BCD set is randomly partitioned into two parts with a different ratio, one for a testing set and another one for a training set. For a given testing/training ratio, 500 tests are performed, and the average and the standard deviation of the prediction success rate on the testing set is recorded. The standard deviation is shown in the parentheses.

[b] The average and the standard deviation of the number of predictions made by the *Z*-score threshold approach. The *z*-score threshold value is shown, which makes the number of prediction by the *Z*-score threshold approach closest to that of the consensus approach.

[c] The average number and the standard deviation of the predictions made both by the consensus approach and the *Z*-score threshold approach.

factor (i.e. consensus between threading$_{\text{const\_weight}}$ and the Smallest *Z*-score Method), that prediction is expected to be highly accurate. On the other hand, disagreement of a predicted template for a target by threading$_{\text{const\_weight}}$ and the Top Rank Frequency Method (or either of the Smallest *Z*-score Method and the Largest *Z*-score Gap Method) strongly indicates that the selected template is wrong. Indeed, when the two methods disagree, the success rate drops to 17.03, 24.25, and 22.33%, when threading$_{\text{const\_weight}}$ disagrees with the Top Rank Frequency Method, the Smallest *Z*-score Method, and the Largest *Z*-score Gap Method, respectively. Therefore, our three methods can be used to increase the confidence of the prediction of threading$_{\text{const\_weight}}$ by combining with it.

One might ask how does the confidence established by the consensus approach above (Fig. 10) compare with the conventional way of using a *Z*-score threshold in threading$_{\text{const\_weight}}$.[1,2] In threading$_{\text{const\_weight}}$, usually a prediction with a *Z*-score which is more significant than a predefined threshold value is considered to be confident. To answer the question, in Table III we further compare the confidence established by consensus between threading$_{\text{const\_weight}}$ and the Top Rank Frequency Method (we call the consensus approach here) with the confidence determined by a *Z*-score threshold value (we call the *Z*-score threshold approach). Here we consider the consensus between the Top Rank Frequency Method and threading$_{\text{const\_weight}}$ because consensus with the other two methods (the Smallest *Z*-score Method and the Largest *Z*-score Gap Method) show even higher success rate (Fig. 10). The *Z*-score threshold value is determined so that threading$_{\text{const\_weight}}$ yields the same number of predictions above the threshold as the consensus between the two methods makes (the third column from the left in Table III). There is a large overlap between predictions made by the consensus approach and the *Z*-score threshold approach (on average 91.6%, the middle column named Overlap). However, the success rate of the consensus approach is slightly higher than the *Z*-score threshold approach (the two rightmost columns).

Figure 11 illustrates the difference of predictions made by the consensus approach and the *Z*-score threshold approach. The *Z*-score threshold approach considers all predictions made with a *Z*-score at the threshold value or smaller (−4.36, indicated by an arrow) as confident predictions, while the consensus approach selects predictions which are consensus between the Top Rank Frequency Method and the threading$_{\text{const\_weight}}$, regardless of the *Z*-score assigned to the predictions. In Figure 11, the predictions made by the consensus approach are shown as Cons_Correct and Cons_Wrong in the histogram. The predictions by the two approaches in the *Z*-score range of below −5.36 are almost identical. Focusing in the *Z*-score range larger than −5.36, the *Z*-score threshold approach makes 82.65 correct predictions and 73.0 wrong predictions on average between the *Z*-score range of −5.36 and the *Z*-score threshold value, −4.36. On the other hand, the consensus approach makes 88.2 correct predictions and 69.9 wrong predictions above the *Z*-score range of −5.36, selecting more correct predictions than the wrong predictions. Thus, the consensus approach performs better than the conventional way of using a constant *Z*-score threshold value in choosing more correct predictions with less wrong predictions.

DISCUSSION

Optimizing weighting factors for a linear combination of terms in a scoring function is a key for success in

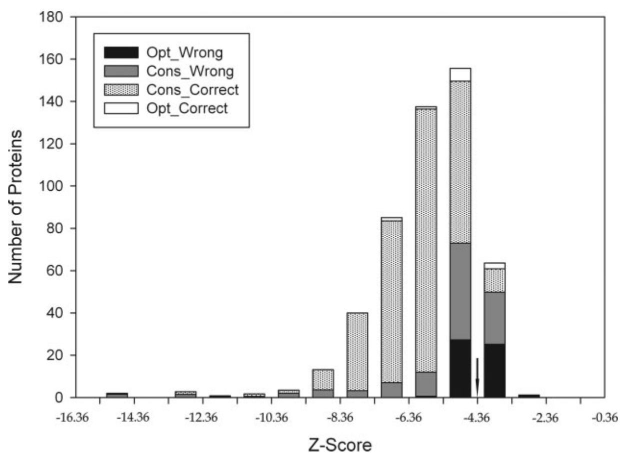


Figure 11

The correct and wrong predictions among the predictions with a confidence assigned by the consensus approach and the *Z*-score threshold approach. The *x*-axis shows the *Z*-score of predictions computed by threading$_{\text{const\_weight}}$. The *y*-axis is the average number of predictions made with the specified *Z*-score among 500 tests with a testing/training data set ratio of 507/510. Opt_Wrong, wrong predictions made by threading$_{\text{const\_weight}}$ which are not selected by the consensus approach; Cons_Wrong, wrong predictions selected by the consensus approach. Therefore, Opt_Wrong + Cons_Wrong is the total wrong predictions by threading$_{\text{const\_weight}}$. Opt_Correct, Correct predictions made by threading$_{\text{const\_weight}}$ but not selected by the consensus approach; Cons_Correct, correct predictions selected by the consensus approach. Therefore, Opt_Correct + Cons_Correct is the total correct predictions by threading$_{\text{const\_weight}}$. The arrow at −4.36 indicates the *Z*-score threshold value for threading$_{\text{const\_weight}}$ (see Table III). Therefore, threading$_{\text{const\_weight}}$ alone will predict all the predictions below the *Z*-score of −4.36 as confident predictions. On the other hand, the consensus approach predicts proteins in Cons_Correct and Cons_Wrong as confident predictions.

developing a threading algorithm. Unlike the conventional threading which uses a constant set of weighting factors that are determined to yield the highest success rate on a training dataset, we have proposed three novel methods which neither rely on a training dataset nor use constant weighting factors: the first method, named the Top Rank Frequency Method, makes a vote to top scoring template structures for a given target sequence at each time it changes the weighting factors of a scoring function used. Then the template which is voted the most time is selected. This method is considered to belong to the class of consensus methods or ensemble approach,[65–71] which combine results obtained by different methods or different parameters. The other two methods, the Smallest *Z*-score Method and the Largest *Z*-score Gap Method select a template which is assigned the most significant *Z*-score (the Smallest *Z*-score Method) or the largest *Z*-score gap between the best and the second best scoring template (the Largest *Z*-score Gap Method) by any of the weighting factors used. As a result, in all of the three methods, every target sequence can potentially use its specific tailor-made weighting factors. Additionally, we have proposed to predict the target-specific weighting factors by neural network from features of a target sequence. All of the four methods aim to break the convention of current threading methods which routinely use constant weighting factors optimized on a training dataset.

In this study, first we have demonstrated that each target protein has its own specific preferable weighting factor which is needed to recognize its native structure in a template library (Table II, Figs. 4, 6, and 8). It should be noted that if a scoring function perfectly describes the free energy of protein structures, of course a single set of parameters should work for any proteins. However, in the current realistic situation that the accurate form of the scoring function is still unclear and each scoring term is imperfect, in practice target-specific weighting factors will give better performance over constant weighting factors in threading. Indeed it is shown that our four novel methods which circumvent weighting factor optimization using a training set perform as good as threading$_{\text{const\_weight}}$ (Table II, Figs. 7 and 9). The detailed comparison between the prediction success rate of the Top Rank Frequency Method and threading$_{\text{const\_weight}}$ has shown no statistical difference in general (Figs. 7 and 9). Moreover, when the size of the training data set is small, often the Top Rank Frequency Method performed better than threading$_{\text{const\_weight}}$ (Fig. 7). Finally, we have shown that the consensus approach between threading$_{\text{const\_weight}}$ and the Top Rank Frequency Method (or the Smallest *Z*-score Method, the Largest *Z*-score Gap Method) has a significantly improved success rate (Fig. 10, Table III).

At this juncture it would be interesting to consider the underlined mechanism why our four novel methods work. The Top Rank Frequency Method presumes that a protein sequence and its native structure fits to each other so much better compared to alternative templates that a certain range of fluctuation of parameters does not affect much to the recognition of the native structure by the target sequence. Considering the energy landscape of protein folding,[72] this may correspond to searching a template structure which locates at the bottom of a wide deep basin which have a large mouth to attract conformations with slightly different energies.[73] On the other hand, the Smallest *Z*-score Method and the Largest *Z*-score Gap Method aim to pinpoint a weighting factor for a target sequence which places a template structure at the bottom of a significantly deep well in the energy landscape. Here not the width but only the relative depth of the deepest well in the landscape is considered. In a sense, our three methods use a reversed logic from the usual threading procedure: in the usual threading, a template structure is considered to be the native for a given target sequence if the structure has a significantly better score compared to the other structures in a template library. Oppositely, our three methods design the energy landscape (i.e. the scoring function) so that a template structure (hopefully the native structure of a target protein) has a local and global landscape which is characteristic to that of the native structure. The two characteristics of the local/global landscape of a native structure for a target protein, namely, the width and the depth, are not necessarily mutually exclusive, as also implied in our results in Table II. Therefore, combining these three methods is expected to improve the success rate, for example by taking consensus of these methods, or by developing a unified score which explicitly combine the frequency that a template is ranked within top k rank, the best Z-score, and the Z-score gap. Another idea for further improvement is to consider similarity between template structures. This may be especially well suited for the top rank frequency method. If the most voted template and the second voted template have significant structure similarity (e.g. if both of them belong to the same SCOP fold), it would be more probable that the identified template is correct for the target sequence. In the actual implementation, for example, when a template is ranked within a certain rank, a partial vote will be given to similar templates, which reflects their structural similarity.

The fourth method presented, prediction of a target-specific weighting factor by machine learning techniques is another promising direction to be investigated. We have shown the first proof of concept of this method, which can be expected to make significant improvement with more suitable input parameters of structure features, including predicted secondary structures, and the contact order of a target sequence.

In this study we used a gapless threading as the model system of threading. In a real application of threading for structure prediction, threading alignment should allow gaps, which introduce another complexity. At this point it is not clear whether the Top Rank Frequency Method, the Smallest Z-score Method, and the Largest Z-score Gap Method perform well in the threading with gaps. Gap penalties could be handled as parameters together with weighting factors, whose possible values can be exhaustively explored within a certain manageable range with a certain step size. However, introducing gaps not merely adds additional parameters to be explored, but also makes it impossible to precompute the score of each term for a given pair of a target and a template as we have done in this study. Rather, threading should be performed for every set of weighting factors and gap penalties, since the score of each term usually depends on the target-template alignment. Thus, the number of threading computation for a target sequence to be performed will now become $N \times M$ with N being the number of templates and M being the possible sets of parameters, which can be a huge impractical number especially when more scoring terms are used. Here we list several ideas to reduce the number of threading computation for a target. First of all, our threading approaches can be used only when a conventional threading with constant weighting factors does not identify obvious templates with a significant Z-score (e.g. a Z-score of -6.36 in Fig. 11). Second, obviously irrelevant templates for a target can be removed from the library (thus reducing N). For example, template structures which belong to a different secondary structure class from the predicted class of the target can be eliminated in a search. Third, probably we can preselect "popular" weighting factors sets to explore, rather than using all possible weighting factor sets (thus reducing M). Those popular sets could be identified by testing the methods on a representative set of target sequences. Last, we would like to mention that grid computing system is well suited for our three methods, since each threading can be performed independently and in a parallel fashion. Application of our methods using some of the ideas above is left as an intriguing future work. It is also interesting to apply our three methods for reranking of protein docking conformations,[23] since it does require gaps.

In summary, we proposed strategies to break the convention of threading which uses constant parameters trained based on a limited size of existing data set. These strategies are aimed to be able to predict the structure of unforeseen target sequences, which may have peculiar preference of parameters. Threading needs a breakthrough to expand its applicability to be able to detect not merely structure of distant homologs but to evolutionary unrelated structures, and this work is intended to pursue toward that direction.

REFERENCES

1. Skolnick J, Kihara D. Defrosting the frozen approximation: PROSPECTOR—a new approach to threading. Proteins 2001;42:319–331.
2. Skolnick J, Kihara D, Zhang Y. Development and large scale benchmark testing of the PROSPECTOR 3.0 threading algorithm. Proteins 2004;56:502–518.
3. Bowie JU, Luthy R, Eisenberg D. A method to identify protein sequences that fold into a known three-dimensional structure. Science 1991;253:164–170.
4. Kihara D, Skolnick J. The PDB is a covering set of small protein structures. J Mol Biol 2003;334:793–802.
5. Orengo CA, Jones DT, Thornton JM. Protein superfamilies and domain superfolds. Nature 1994;372:631–634.
6. Kryshtafovych A, Venclovas C, Fidelis K, Moult J. Progress over the first decade of CASP experiments. Proteins 2005;61 (Suppl):7225–7236.
7. Wang G, Jin Y, Dunbrack RL, Jr. Assessment of fold recognition predictions in CASP6. Proteins 2005;61 (Suppl):746–766.
8. Rychlewski L, Jaroszewski L, Li W, Godzik A. Comparison of sequence profiles. Strategies for structural predictions using sequence information. Protein Sci 2000;9:232–241.
9. Wang G, Dunbrack RL, Jr. Scoring profile-to-profile sequence alignments. Protein Sci 2004;13:1612–1626.
10. Karchin R, Cline M, Mandel-Gutfreund Y, Karplus K. Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. Proteins 2003;51:504–514.

11. Karplus K, Barrett C, Hughey R. Hidden Markov models for detecting remote protein homologies. Bioinformatics 1998;14:846–856.
12. Dunbrack RL, Jr. Sequence comparison and protein structure prediction. Curr Opin Struct Biol 2006;16:374–384.
13. Zhou H, Zhou Y. Single-body residue-level knowledge-based energy score combined with sequence-profile and secondary structure information for fold recognition. Proteins 2004;55:1005–1013.
14. Matsuo Y, Nishikawa K. Protein structural similarities predicted by a sequence-structure compatibility method. Protein Sci 1994;3: 2055–2063.
15. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 1999;292:195–202.
16. Rost B, Schneider R, Sander C. Protein fold recognition by prediction-based threading. J Mol Biol 1997;270:471–480.
17. Skolnick J, Jaroszewski L, Kolinski A, Godzik A. Derivation and testing of pair potentials for protein folding. When is the quasichemical approximation correct? Protein Sci 1997;6:676–688.
18. Sippl MJ. Knowledge-based potentials for proteins. Curr Opin Struct Biol 1995;5:229–235.
19. Kihara D, Lu H, Kolinski A, Skolnick J. TOUCHSTONE: an *ab initio* protein structure prediction method that uses threading-based tertiary restraints. Proc Natl Acad Sci USA 2001;98:10125–10130.
20. Kmiecik S, Gront D, Kolinski A. Towards the high-resolution protein structure prediction. Fast refinement of reduced models with all-atom force field. BMC Struct Biol 2007;7:43.
21. Zhou H, Skolnick J. *Ab initio* protein structure prediction using chunk-TASSER. Biophys J 2007;93:1510–1518.
22. Halperin I, Ma B, Wolfson H, Nussinov R. Principles of docking: an overview of search algorithms and a guide to scoring functions. Proteins 2002;47:409–443.
23. Pierce B, Weng Z. ZRANK: reranking protein docking predictions with an optimized energy function. Proteins 2007;67:1078–1086.
24. Meiler J, Baker D. ROSETTALIGAND: protein-small molecule docking with full side-chain flexibility. Proteins 2006;65:538–548.
25. Sousa SF, Fernandes PA, Ramos MJ. Protein-ligand docking: current status and future challenges. Proteins 2006;65:15–26.
26. Wu Y, Lu M, Chen M, Li J, Ma J. OPUS-Ca: a knowledge-based potential function requiring only Calpha positions. Protein Sci 2007;16:1449–1463.
27. Tan YH, Huang H, Kihara D. Statistical potential-based amino acid similarity matrices for aligning distantly related protein sequences. Proteins 2006;64:587–600.
28. Akutsu T, Tashimo H. Linear programming based approach to the derivation of a contact potential for protein threading. Pac Symp Biocomput 1998;413–424.
29. Zien A, Zimmer R, Lengauer T. A simple iterative approach to parameter optimization. J Comput Biol 2000;7:483–501.
30. Antes I, Merkwirth C, Lengauer T. POEM: parameter optimization using ensemble methods: application to target specific scoring functions. J Chem Inf Model 2005;45:1291–1302.
31. Rosen JB, Phillips AT, Oh SY, Dill KA. A method for parameter optimization in computational biology. Biophys J 2000;79:2818–2824.
32. Chiu TL, Goldstein RA. Optimizing energy potentials for success in protein tertiary structure prediction. Fold Des 1998;3:223–228.
33. Mirny LA, Shakhnovich EI. How to derive a protein folding potential? A new approach to an old problem. J Mol Biol 1996;264:1164–1179.
34. Maiorov VN, Crippen GM. Contact potential that recognizes the correct folding of globular proteins. J Mol Biol 1992;227:876–888.
35. Hao MH, Scheraga HA. How optimization of potential functions affects protein folding. Proc Natl Acad Sci USA 1996;93:4984–4989.
36. Noguchi T, Akiyama Y. PDB-REPRDB: a database of representative protein chains from the Protein Data Bank (PDB) in 2003. Nucleic Acids Res 2003;31:492–493.
37. Silverman BW. Density estimation for statistics and data analysis, 1st ed. New York: Chapman & Hall/CRC; 1986.
38. Betancourt MR, Thirumalai D. Pair potentials for protein folding: choice of reference states and sensitivity of predicted native states to variations in the interaction schemes. Protein Sci 1999;8:361–369.
39. Miyazawa S, Jernigan RL. An empirical energy potential with a reference state for protein fold and sequence recognition. Proteins 1999;36:357–369.
40. Zhou H, Zhou Y. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. Protein Sci 2002;11:2714–2726.
41. Skolnick J. In quest of an empirical potential for protein structure prediction. Curr Opin Struct Biol 2006;16:166–171.
42. Huang ES, Samudrala R, Ponder JW. Ab initio fold prediction of small helical proteins using distance geometry and knowledge-based scoring functions. J Mol Biol 1999;290:267–281.
43. Shen MY, Sali A. Statistical potential for assessment and prediction of protein structures. Protein Sci 2006;15:2507–2524.
44. Benkert P, Tosatto SC, Schomburg D. QMEAN: a comprehensive scoring function for model quality assessment. Proteins 2008;71: 261–277.
45. Colovos C, Yeates TO. Verification of protein structures: patterns of nonbonded atomic interactions. Protein Sci 1993;2:1511–1519.
46. Melo F, Feytmans E. Novel knowledge-based mean force potential at atomic level. J Mol Biol 1997;267:207–222.
47. Zhang J, Chen R, Liang J. Empirical potential function for simplified protein models: combining contact and local sequence-structure descriptors. Proteins 2006;63:949–960.
48. Gront D, Kolinski A. A new approach to prediction of short-range conformational propensities in proteins. Bioinformatics 2005;21:981–987.
49. Lu H, Skolnick J. A distance-dependent atomic knowledge-based potential for improved protein structure selection. Proteins 2001;44: 223–232.
50. Betancourt MR. A reduced protein model with accurate native-structure identification ability. Proteins 2003;53:889–907.
51. Tobi D, Elber R. Distance-dependent, pair potential for protein folding: results from linear optimization. Proteins 2000;41:40–46.
52. Reva BA, Finkelstein AV, Skolnick J. What is the probability of a chance prediction of a protein structure with an rmsd of 6 A? Fold Des 1998;3:141–147.
53. Sali A, Shakhnovich E, Karplus M. How does a protein fold? Nature 1994;369:248–251.
54. Ivankov DN, Finkelstein AV. Prediction of protein folding rates from the amino acid sequence-predicted secondary structure. Proc Natl Acad Sci USA 2004;101:8942–8944.
55. Nakashima H, Nishikawa K, Ooi T. The folding type of a protein is relevant to the amino acid composition. J Biochem (Tokyo) 1986;99:153–162.
56. Klein P, DeLisi C. Prediction of protein structural class from the amino acid sequence. Biopolymers 1986;25:1659–1672.
57. Chou KC. Progress in protein structural class prediction and its impact to bioinformatics and proteomics. Curr Protein Pept Sci 2005; 6:423–436.
58. Adamczak R, Porollo A, Meller J. Combining prediction of secondary structure and solvent accessibility in proteins. Proteins 2005;59: 467–475.
59. Bharath R, Bharath R, Drosen J. Neural network computing. New York: Windcrest/McGraw-Hill; 1994.
60. Betancourt MR, Skolnick J. Local propensities and statistical potentials of backbone dihedral angles in proteins. J Mol Biol 2004;342: 635–649.
61. Fleming PJ, Gong H, Rose GD. Secondary structure determines protein topology. Protein Sci 2006;15:1829–1834.
62. Kihara D. The effect of long-range interactions on the secondary structure formation of proteins. Protein Sci 2005;14:1955–1963.
63. Devore JL. Inferences based on two samples; probability and statistics for engineering and the sciences, 6th ed., Chapter 9. Minnesota: Brooks/Cole Publishing Company; 2004. pp 354–401.

64. Lo Conte L, Brenner SE, Hubbard TJ, Chothia C, Murzin AG. SCOP database in 2002: refinements accommodate structural genomics. Nucleic Acids Res 2002;30:264–267.
65. Chivian D, Baker D. Homology modeling using parametric alignment ensemble generation with consensus and energy-based model selection. Nucleic Acids Res 2006;34:e112.
66. Fischer D. 3D-SHOTGUN: a novel, cooperative, fold-recognition meta-predictor. Proteins 2003;51:434–441.
67. Ginalski K, Elofsson A, Fischer D, Rychlewski L. 3D-Jury: a simple approach to improve protein structure predictions. Bioinformatics 2003;19:1015–1018.
68. Hu J, Yang YD, Kihara D. EMD: an ensemble algorithm for discovering regulatory motifs in DNA sequences. BMC Bioinformatics 2006;7:342.
69. Contreras-Moreira B, Fitzjohn PW, Bates PA. *In silico* protein recombination: enhancing template and sequence alignment selection for comparative protein modelling. J Mol Biol 2003;328:593–608.
70. Wallner B, Elofsson A. Pcons5: combining consensus, structural evaluation and fold recognition scores. Bioinformatics 2005;21: 4248–4254.
71. Notredame C, Higgins DG, Heringa J. T-Coffee: a novel method for fast and accurate multiple sequence alignment. J Mol Biol 2000;302: 205–217.
72. Dill KA, Chan HS. From Levinthal to pathways to funnels. Nat Struct Biol 1997;4:10–19.
73. Nakamura HK, Sasai M. Population analyses of kinetic partitioning in protein folding. Proteins 2001;43:280–291.