Statistica Sinica **12**(2002), 159-177

# A PRACTICAL METHOD FOR INTERPRETATION OF THREADING SCORES: AN APPLICATION OF NEURAL NETWORK

Ying Xu, Dong Xu and Victor Olman

*Oak Ridge National Laboratory*

*Abstract:* Protein threading has become a popular technique for protein fold recognition and structure prediction. However it remains a challenging and unsolved problem to assess the significance or reliability of a threading prediction result. The lack of an effective mechanism for such an assessment has greatly limited further applications of threading on a genome-scale. We have developed a practical method for assessing the reliability of a threading result, using a neural network approach. As a key goal of threading is to separate true sequence-fold pairs (a pair of proteins that share the same structural fold) from false sequence-fold pairs, we have examined the distribution of true pairs against the many times more false pairs in the parameter space, and discovered that the vast majority of the true pairs fall into a continuous region without any false ones, providing the basis for pattern recognition using a neural network. We have trained a neural network trying to capture the shape of this "true" region. Our preli minary results are quite encouraging and show that our approach is more effective in assessing the prediction reliability than another neural network-based approach employed in the popular threading program GenThreader. This preliminary study also indicates that our current neural network is too simple to accurately capture the overall shape of the true region, and points to directions for further investigation on this highly important and challenging problem. This neural network-based assessment capability has been implemented in our threading program PROSPECT and used during the CASP4 predictions. Our successful performance in CASP4 suggests that even though this trained neural network is far from being perfect, it is fairly effective.

*Key words and phrases:* Confidence assessment, fold recognition, neural network, protein structure prediction, threading.

## 1. Introduction

A protein consists of a string of amino acids (of twenty different types), and folds into a unique, stable three dimensional (3D) structure in its native state. Though a protein structure may have some dynamic motion in solvent, its movement is generally rather small, and hence a protein structure can be considered as a static geometric object. The protein folding problem can be defined as determining or predicting the 3D structure of a protein from its amino acid sequence.

This problem is of central importance to contemporary molecular biology as the 3D structures of proteins not only determine their biological functions, but also provide a key to design drugs to target particular proteins.

Traditionally, protein structures are solved mainly by experimental methods like X-ray crystallography or NMR. It typically takes months and possibly years to solve one structure. The completion of over 40 genome projects has significantly increased the demand for a higher rate of protein structure determination – with tens of thousands of genes, which encode protein sequences, having been determined. Traditional experimental methods for protein structure determination clearly cannot keep up with the pace at which protein sequences are being generated.

Computational methods represent a promising alternative for protein structure determination. Though their prediction accuracy is not quite on the level of experimental structures yet, the prediction capability of computational methods have been clearly demonstrated in the bi-annual community-wide experiments on the Critical Assessment of Techniques for Protein Structure Prediction (CASP) (CASP (1995), CASP (1997), CASP (1999), CASP (2001)). In each CASP experiment, predictors are given a list of protein sequences whose structures have been solved experimentally (or are expected to be solved during the CASP prediction season) but unpublished. The prediction teams submit their predictions before an expiration date for each prediction target. Their prediction results are then evaluated against the experimental structures at the end of the prediction season. The success of CASP predictions has demonstrated that many predicted structures are accurate enough to make some functional inferences. It is generally expected that a vast majority of the proteins, identified by the genome projects, will be structurally predicted through computational approaches rather than experimental ones within the next ten years.

The protein folding problem can be considered as an optimization problem from a mathematical point of view. It is generally assumed that the native structure of a protein has the conformation corresponding to the lowest free energy. Theoretically if one could model *all* the energy accurately and solve the optimization problem, a protein structure could be predicted as accurately as experiments can measure. However, this is unrealistic. Though all the interactions which govern protein folding can be described by known physical principles, the current computing capability is many orders of magnitude too low to compute the interactions using quantum mechanics because of the enormous amount of computation involved. The folding problem can be simplified through using semi-empirical physical energies, e.g., modeling a chemical bond as a harmonic spring and electrostatics energy with an inverse-square potential as a function of the distance between two charged atoms. A good example of such an energy function is the

CHARMM potential (Brooks et al. (1983)). Some success has been reported in predicting structures of protein segments (about 10 amino acids) through minimizing semi-empirical energies (Li and Scheraga (1987), Pedersen and Moult (1997)). Nevertheless the computation is still too heavy to fold even a small full protein (with less than 100 amino acids) using this type of energy function. To further simplify the problem to make it practically computable, people have used knowledge-based energy function (Skolnick and Kolinski (1991), Goldstein, Luthey-Schulte and Wolynes (1992)), which is derived from the observed spatial relationship of different types of amino acids in the solved structures, e.g., how often a known structure may have an alanine and a proline at certain distance under a particular solvent accessibility. This type of residue-based energy function has significantly decreased the complexity of the folding problem, and it is often good enough to define the overall structure at the level of amino acid rather than atomic detail. Recent study has shown that knowledge-based energy by and large captures the detailed atomic potentials (Mohanty et al. (1999)). However, the folding problem is still not computationally tractable for finding the globally optimal solution, even with this level of simplification. Though some prediction success has been reported with this type of energy function, successful methods on a more consistent basis are yet to come.

Fortunately, nature provides a break to scientists working on this highly challenging problem. It is found that proteins with no apparent sequence similarity may have similar structural folds (a structural fold is the 3D conformation of a protein's backbone) (Levitt and Chothia (1981), Finkelstein and Ptitsyn (1987)). Figure 1 shows such an example. Recent studies have further indicated that the total number of different structural folds in nature may be quite small (Li, Helling, Tang and Wingreen (1996), Wang (1998)), possibly in the range of a few thousand or even fewer, which is at least two orders of magnitude fewer than the number of known protein sequences. Statistics from PDB (Bernstein et al. (1977)) have shown that 90% of the proteins solved in the past three years share a similar structural fold with previously solved structures. This suggests that, for a majority of the proteins, structure prediction problems (on the amino acid level) can be effectively reduced to the problem of searching for the correct folds among all possible structural folds. Even on the atomic level, the problem is effectively decomposed to three significantly easier problems: (1) a protein fold recognition problem, i.e., to determine which structural fold a protein sequence will fold into; (2) an alignment problem, i.e., to find the optimal way to place the protein sequence onto the identified structural fold; and (3) a modeling problem of sidechain conformations, with a constrained backbone structure. This decomposition has fundamentally changed the paradigm of protein structure prediction, making the problem realistically solvable. NIH recently initiated a new project, called the Structural Genomics Initiative (National Institute of General Medical Sciences (1999)). Its goal is to selectively solve a few thousand structures experimentally to have a "complete" coverage of the fold space, and then to solve the rest of the protein structures computationally.

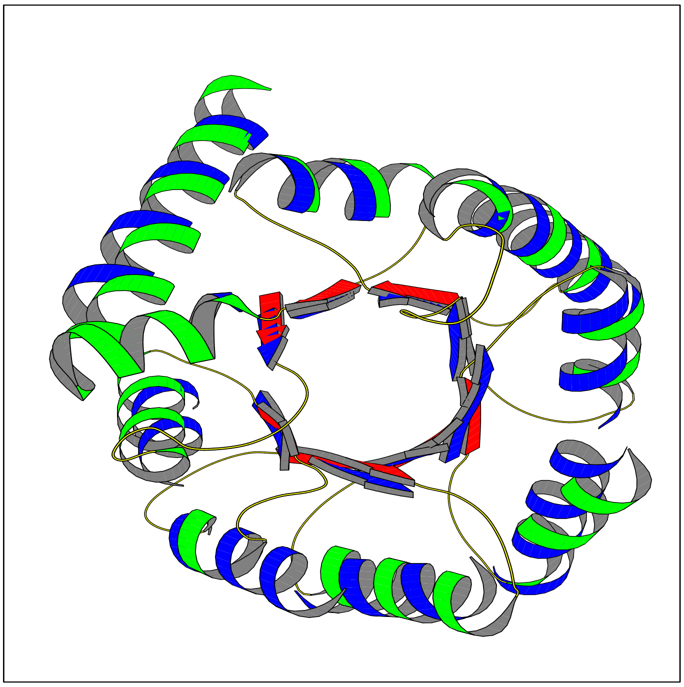


Figure 1. Structure superposition between 1gox (glycolate oxidase, in blue) and 1ak5 (inosine monophosphate dehydrogenase, in green). The sequence identity between the two proteins is 17%, and the root mean square deviation between the two structures is 2.5 Å for the $C_\alpha$ atoms. The structure superposition is obtained through VAST (Gibrat, Madej and Bryant (1996)) and the figure is made using *MOLSCRIPT* (Kraulis (1991)).

In the rest of the paper, we will focus on issues related to the first two problems. For about 30% of new protein sequences, the first two problems can be solved through sequence-sequence comparison (Gerstein (1998)), based on the premise that significant sequence similarity implies significant structural similarity. A more general technique, called *protein threading*, may cover 50-70% of new proteins (Jones (1999)). Threading is particularly good at identifying native-like structures when the query protein sequence and its fold template do not share significant sequence similarity. It achieves its superiority through utilizing not only sequence-level information but also 3D structure information. The threading method achieved its initial success in the pioneering work of Eisenberg and colleagues (Bowie, Luthy and Eisenberg (1991), Luthy, Bowie and Eisenberg (1992)). Now threading has become a popular technique for protein structure prediction through further developments by many other research groups (Sippl and Weitckus (1992), Jones, Taylor and Thornton (1992), Godzik, Skolnick and Kolinski (1992), Bryant and Altschul (1995), Fischer, Rice, Bowie and Eisenberg (1996b), Alexandrov, Nussinov and Zimmer (1996), Xu, Xu and Uberbacher (1998)).

The problem of protein threading can be formulated as follows. Given a query protein sequence **s** of unknown structure, threading searches a structure template library **T** to find the best sequence-structure alignment **s**-**t**, $\mathbf{t} \in \mathbf{T}$, measured by the overall preference of individual residues to their structural environment and of residue-residue contacts (interactions). A threading method typically consists of four components (Smith et al. (1997)): (1) a library **T** of representative 3D protein structures as templates; (2) an energy function for measuring the fitness between a query **s** and a template **t**, where $\mathbf{t} \in \mathbf{T}$; (3) a threading algorithm for searching for the lowest energy among the possible alignments for a given **s**-**t** pair; (4) a criterion for estimating the confidence level of a sequence-structure alignment.

Despite the success of the threading approach, there are still a number of problems to be solved. Among them are (a) the correct folds (the templates that most closely resembles the native fold of the query, or in biological terms, structural *homologs* or *analogs*) are not always ranked as the best; and (b) even when the correct fold is ranked the best, the alignment between the query sequence and the fold template is often not perfect, making further construction of the detailed atomic structure unreliable. These problems have their roots in each of the four threading components. First, the template library is incomplete: though a query protein may have a similar fold in the library, the structural difference between this fold and the query's native fold may be too large to achieve a high ranking. We expect this problem to become less of an issue as more and more new structures are added to the template library through the effort of the NIH Structural Genomics Project. Second, in the algorithmic aspect, existing threading methods often fail to find the alignment with the lowest energy. Significant progress has been made in solving this problem through our recent work, which has resulted in an efficient algorithm that guarantees to find the global energy minimum in threading (Xu, Xu and Uberbacher (1998), Xu and Xu (2000)). The computational efficiency of our algorithm is achieved through fully utilizing the fact that only local pairwise contacts need to be considered for fold recognition and through a discovery that the dominating factor in the computational complexity of our divide-and-conquer threading scheme (Xu, Xu and Uberbacher (1998)) is how complex the overall structure of pairwise contacts is, defined as the topological complexity (Xu and Xu (2000)) of a structural fold, and the topological complexities of all known structural folds are relatively low. Third, there are many energy functions used in different threading programs. Each of them has some discerning power (in terms of recognizing the correct fold and producing the correct alignment), but all have their limitations due to the crudeness of the function form.

The most challenging problem is the assessment of confidence in a threading result. Due to the lack of more effective ways to score threading results, most existing programs use the raw (total) score of threading energy terms to rank sequence-fold alignments. Such a score should apparently be normalized against the lengths of the query sequence and the template protein. In addition, we have observed that other factors could also affect the threading scores. These may include the amino acid composition, the overall geometric shape of a template, etc. For example, we have found that certain templates tend to be ranked high for many query sequences, indicating that the *baseline* threading scores (typical threading score against an arbitrary protein sequence) for these templates are relatively high compared to other templates (and lengths do not seem to be a major contributor here). Due to the nontrivial nature of normalizing the threading scores, there has been no theoretically sound and practically effective method to put all threading scores on the same scale so that score comparisons across different queries and different templates are meaningful.

What has made the sequence-sequence comparison tools like BLAST (Altschul et al. (1997)) or FASTA (Pearson and Lipman (1988)) so popular is their ability to assess the statistical significance of their alignment results. These significance values are estimated based on rigorous and sound statistical models and a biologist can easily interpret the sequence comparison results based on such values. However, there has been no similar model for assessing the statistical significance of a threading result due to a number of complicating factors. This has clearly limited the usage of the threading methods.

We have developed a practical method for assessing the significance of a threading result using a neural network approach. To show our method, we first introduce the mathematical formulation of protein threading. Then we describe our assessment method and give some prediction results related to the confidence assessment. We end with a discussion on future developments.

## 2. Problem Formulation

### 2.1. Mathematical representation of threading problem

A protein threading problem can be defined as to find a sequence-structure alignment that optimizes the sum of three terms: (a) the singleton fitness energy, (b) the pairwise contact energy, plus (c) the alignment gap penalties. A protein 3D structure is defined as a sequence of amino acids and their 3D coordinates, which may consist of three types of secondary structures: $\alpha$-helices, $\beta$-strands, and loops. For convenience, we call the $\alpha$-helices and $\beta$-strands *core* secondary structures. Generally, core secondary structures are well-conserved while loops may not be among the homologous/analogous structures. Hence we consider only core secondary structures in our energy calculation and penalize the length difference between the aligned loop regions (without considering the detailed fitness or contact energy). In our formulation of the threading problem, no alignment gap is allowed in the core secondary structure regions. In modeling the pairwise contact potentials, it is generally believed that only local residue-residue contacts need to be considered for the purpose of threading alignment (Jones, Taylor and Thornton (1992), Alexandrov, Nussinov and Zimmer (1996)). We have used a simple cutoff distance between the $C_\beta$ atoms of two residues to determine if they have enough contact to be considered.

More formally, we define a threading problem as follows. Let $s = s_1s_2\cdots s_n$ be a query protein sequence, and $(t, T)$ be a protein structure template with loops removed (loop lengths are kept), where $t = t_1t_2\cdots t_m$ is a sequence of template positions with an array of physical properties attached to each of them, and $T = T_1, \ldots, T_M$ is the sequence of core secondary structures partitioning $t$. Thus the $T_i$ are contiguous segments of $t$, $T_i$ representing the $i^{th}$ core secondary structure. Let "$\mathit{pairs}(t, T)$" be the set of all pairs of positions in the template $(t, T)$ considered to have pairwise contacts (in our current implementation, a pair of residues are considered to be in contact if their $C_\beta$ atoms are within 8 Å). Further, let "$\mathit{loop}(T_i, T_{i+1})$" represent the length of the loop between $T_i$ and $T_{i+1}$. We use $f(x, y)$ to represent the fitness of aligning (or placing) the amino acid $x$ to the template position $y$, $x \in s$ and $y \in t$, with a collection of attributes that describe the physical properties at position $y$. We use $c(x_1, x_2)$ to denote the contact potential of a pair $(x_1, x_2) \in \mathit{pairs}(t, T)$ and $p(|r_1 - r_2|)$ to represent the penalty for the length difference between a template loop (length $r_1$) and its "aligned" portion (length $r_2$) of the query sequence. The protein threading problem is to find a partition $\{S_1, \ldots, S_{2M+1}\}$ of $s$ such that $\|S_{2i}\| = \|T_i\|$ for all $1 \le i \le M$, and for which the following function is minimized:

$$\sum_{i\in[1,M]} \sum_{j\in[1,\|T_i\|]} f(S_{2i}[j], T_i[j]) + \sum_{(T_i[j], T_{i'}[j'])\in \mathit{pairs}(t,T)} c(S_{2i}[j], S_{2i'}[j']) + \sum_{i\in[2,M]} p(|\, \|S_{2i-1}\| - \mathrm{loop}(T_{i-1}, T_i)\,|), \tag{1}$$

where $X[k]$ represents the $k^{th}$ element of $X$, $\|\cdot\|$ represents the cardinality of a set, and $S_{2i-1}$ represents the loop region between $S_{2i-2}$ and $S_{2i}$, for $2 \le i \le M$. Note here that each $S_i$ is a substring of $s$, some possibly empty.

We have developed an algorithm for solving this optimization problem. It finds the globally optimal threading alignment, and runs efficiently when the cutoff for residue-residue contacts is 7 or 8 Å (Xu, Xu and Uberbacher (1998)). We have implemented it as the computer program PROSPECT (Xu and Xu (2000)).

### 2.2. Energy function

We now outline how the $f$, $c$ and $p$ terms found in expression (1) are calculated. In PROSPECT,

$$f = E_{mutate} + E_{single}, \tag{2}$$

$c$ is also called $E_{pair}$, and $p$ is a linear function which can be expressed for an alignment gap of length $g$ as in (Gonnet, Cohen and Benner (1992)):

$$p(g) = A + B * (g - 1), \tag{3}$$

where $A$ and $B$ are positive constants, and $g > 0$.

The mutation energy $E_{mutate}$ describes the compatibility of substituting one amino acid type by another, and the singleton energy $E_{single}$ represents a residue's preference to its local secondary structure and its preference to being in a certain solvent environment (either exposed to solvent or in the interior of the protein). For $E_{mutate}$, several matrices have been developed based on mutation rates found in sequence databases. The most popular of these are the PAM (Dayhoff (1978)) and BLOSUM (Henikoff and Henikoff (1992)) matrices. The BLOSUM-62 is a widely-used matrix for detecting close homologs, while PAM250 (Gonnet, Cohen and Benner (1992)) is more suitable for identifying remote homologs. Experience has shown that PAM250 is one of the best mutation matrices available for threading (Fischer, Rice, Bowie and Eisenberg (1996b), Abagyan and Batalov (1997)).

Both $E_{single}$ and $E_{pair}$ are typically derived from Boltzmann statistics from a set of non-homologous proteins. The basic idea is that if an amino acid is frequently observed in the interior of protein structures, a favorable energy value will be rewarded when it is aligned to an interior position of a template. We calculate $E_{single}$ as $E_{single} = \sum_i e_{single}(i, ss_i, sol_i)$, where $e_{single}(i, ss, sol)$ represents the energy or preference for a particular combination of amino acid type $i$, secondary structure type $ss$, and solvent accessibility type $sol$. It is expressed as

$$e_{single}(i, ss, sol) \; = \; - \log \frac{N(i, ss, sol)}{N_E(i, ss, sol)} \;\; , \tag{4}$$

log is the natural logarithm, $N(i, ss, sol)$ is the number of amino acids of type $i$ in the environment defined by $ss$ and $sol$, as counted from the FSSP database (Holm and Sander (1996)), $N_E(i, ss, sol)$ is the estimated number of amino acids of type $i$ in $ss$ and $sol$ assuming $i$, $ss$, and $sol$ are independent. $N_E(i, ss, sol)$ is calculated as

$$N_E(i, ss, sol) \; = \; \frac{N(i)\, N(ss)\, N(sol)}{N^2} \;\; , \tag{5}$$

where $N(i)$ is the number of amino acids of type $i$, $N(ss)$ is the number of residues in secondary structures of type $ss$, $N(sol)$ is the number of residues with solvent accessibility $sol$, and $N$ is the total number of residues in the non-homologous protein database.

Similarly, $E_{pair}$ is calculated as $E_{pair} = \sum_{i \leq j} e_{pair}(i,j)$, where $e_{pair}(i,j)$ is derived from the frequency of the inter-residue pairs. That is,

$$e_{pair}(i,j) \;=\; -\log \frac{M(i,j)}{M_E(i,j)} \;,\quad M_E(i,j) \;=\; \frac{M(i)\,M(j)}{M} \;, \tag{6}$$

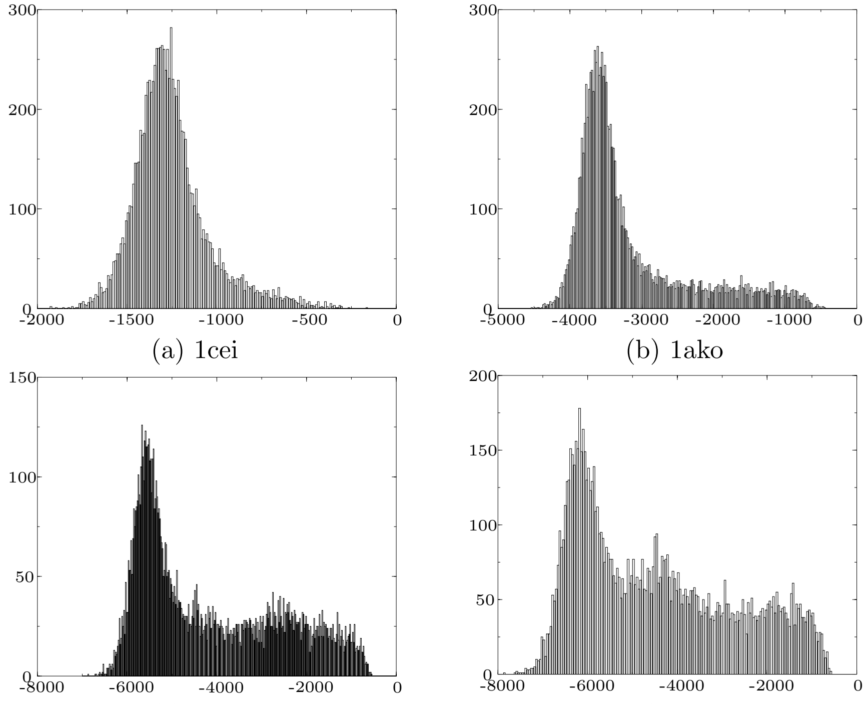
where $M(i,j)$ is the number of pairs of residues of types $i$ and $j$ in the database (within the cutoff), and $M_E(i,j)$ is the estimated number of $i$-$j$ pairs assuming that residues of types $i$ and $j$ form a pair without any *apriori* preference. Then $M(k) = \sum_s M(k,s)$ is the number of all pairs which consist of residue of type $k$ and $M = \sum_{ij} M(i,j)$ is the total number of pairs in the database. In the calculation, the number of pairs of residues of types $i$ and $j$ are partitioned equally in $M(i,j)$ and $M(j,i)$, $M(i,j) = M(j,i)$. In PROSPECT, the cutoff distance for $e_{pair}(i,j)$ is 7.0 Å between $C_\beta$ atoms, which accounts for most of the important inter-residue interactions (Jones, Taylor and Thornton (1992), Alexandrov, Nussinov and Zimmer (1996)). We only consider the residue pairs that are separated by at least 3 amino acids in the protein sequence. Pairs separated by one or two amino acids in the protein sequence represent local interactions, which are less important in determining an overall fold.

### 2.3. Assessment of threading results

For applications, a threading program needs to calculate an optimal alignment between a query sequence and each structure in the template library. Then a decision needs to be made as to which sequence-structure alignment likely gives the correct fold recognition (and the "correct" alignment). This is a highly challenging and unsolved problem. In principle, one could build a probabilistic model of the knowledge-based threading energy that accounts for different factors and the correlations among these factors. Such a model could directly give a probability (or reliability) of a particular sequence-structure alignment being correct. There have been a number of such attempts to do this. An early one was to use *z-score* (Flockner et al. (1995)). For the energy $E$ resulting from a particular alignment, the $z$-score of $E$ is

$$z \;=\; \frac{E - \bar{E}}{\sigma} \;, \tag{7}$$

where $\bar{E}$ and $\sigma$ are, respectively, the average and standard deviation of the energy distribution resulting from the same alignment after re-shuffling the amino acids of the query sequence. However it has been shown that the $z$-score is not effective (Marchler-Bauer and Bryant (1997)). There have also been attempts to use the P-value scheme (Karlin, Dembo and Kawabata (1990), Karlin and Altschul (1990)) to build such a model. A P-value estimates the probability of having alignment scores between two random sequences higher than a particular value, and has been successfully applied to sequence alignment, thanks to Karlin's seminal work on a rigorous model for gapless alignments (Karlin, Dembo and Kawabata (1990), Karlin and Altschul (1990)). Due to the lack of a rigorous model for threading, the P-values are typically estimated through compiling a "large" number of threading scores between a query sequence and a template after randomly shuffling its residues (Bryant and Altschul (1995)). While some usefulness of the estimated P-value has been demonstrated, the problem of developing a rigorous and effective scheme for threading remains a challenging open problem.



(a) 1cei

(b) 1ako

(c) 1aszb

(d) 1lci

Figure 2. Threading score distributions of four templates against 10,000 sequences randomly selected from the non-redundant protein sequences database in PIR (Barker et al. (1999)). The x-axis is the threading score axis and the y-axis represents the occurrences. (a) 1cei (85 amino acids). (b) 1ako (268 amino acids). (c) 1aszb (490 amino acids). (d) 1lci (523 amino acids).

To further illustrate the necessity of “normalizing” threading scores, we show the scoring distributions of four template structures against the 10,000 protein sequences in Figure 2. Clearly a particular threading score, say -2000, may have a quite different meaning for different templates. Based on our current understanding, the contributing factors to variations in score distributions not only include sequence/template lengths, but also such characteristics as amino acid compositions, structural features, etc. It is clearly very challenging to put all these characteristics into one mathematical model in order to rigorously estimate the significance or reliability of a threading result.

## 3. Results

We have developed a practical method for assessing the significance of a threading result using a neural network approach. The basic idea can be described as follows. We selected a large set of query sequences and threaded them against our template library, using PROSPECT. The threading result is a set of 36,657 sequence-structure alignments, among which 1469 are true pairs and the rest false. A true pair is defined as a pair of proteins that belong to the same fold in the FSSP database (Holm and Sander (1996)), otherwise a pair is considered false. A key distinguishing characteristic between true and false pairs is that the number of structurally alignable residues (Alexandrov (1996)) in a true pair is generally significantly higher than the number of structurally alignable residues in a false pair. Table 1 summarizes the relative frequencies of true pairs among all 36,657 pairs with numbers of structurally alignable residues in a particular range:

Table 1. Relative frequency of true pairs vs. number of structurally alignables.

| number of alignables | 0-9 | 10-19 | 20-29 | 30-39 | 40-49 | 50-59 | 60-69 | 70-79 | 80-89 | 90- |
|---|---|---|---|---|---|---|---|---|---|---|
| relative frequency of true pairs | 0.000 | 0.001 | 0.003 | 0.020 | 0.150 | 0.500 | 0.750 | 0.900 | 0.990 | 1.000 |

If the number of structurally alignable residues in a true query-template pair is $x$, a perfect threading alignment should have these $x$ pairs of residues aligned accordingly. Due to the crudeness of existing residue-based energy functions, our threading program generally does not consistently produce threading alignments (for true pairs) with 100% accuracy. For a partially correct alignment, we can assume that the score contribution from incorrectly aligned residues is consistent with the background scores for this template, i.e., for scores of false pairs involving this template. Hence a threading score for a true pair should essentially reflect the number of structurally alignable residues that get aligned correctly, after deducting the background scores for this particular template. The first goal of our study is to find a function which can approximately map a threading score between a pair of proteins, along with various other contributing factors, to the number of structurally alignable residues between the pair. Other contributing factors should include enough information to help the mapping function to deduct background scores. Here we have included as candidates (1) the lengths of the query sequence and the template sequence, (2) information about the template's score distribution (including the extreme values, the average and the standard deviation of the distribution), (3) the sequence identity between a query and the template sequence, and (4) the number of core secondary structures in the template and the total length of its core residues.

We used a neural network approach (Hertz, Krogh and Palmer (1991)) to construct the mapping. We trained the network on a data set consisting of 50% of the 36,657 pairs, and tested it on the remaining 50% of the data points. The input vector to the neural network consists of the threading scores along with parameters listed in (1) – (4), and the desired output value is the the number of structurally alignable residues between each pair (in fact, we use frequencies instead of the number of alignables – see Table 1).

Before we present the training details and results, we first describe some preliminary results of our study on the distribution of the true pairs among all pairs in the parameter space. Only if the true points occupy a region or regions without significant numbers of false points, can we expect the neural network training to lead to good separation between true and false pairs.

### 3.1. Properties of true-pairs

We represent each query-template pair by a vector of 10 parameters, including its threading scores by PROSPECT and various parameters describing various physical and geometric features of the template and the length of the query, as listed in (1) – (4). We want to investigate the regions occupied only by true pairs in the 10-D space. These are 36,657 vectors, 1469 of which represent true pairs and the rest are all false pairs. Our initial attempt involves trying to identify a small number of maximal spheres (or other convex sets), which consist of only true pairs. A maximal sphere is defined as a sphere that contains only true pairs and cannot be enlarged. The largest such sphere consists of 171 true pairs. Further investigation leads to a number of such spheres consisting of at least 50 true pairs. We found that some of these spheres overlap with each other. As we lower the threshold on the size of a sphere to 2 true pairs, we found that 1425 out of 1469 true pairs are contained in a series of overlapping spheres, and all these spheres are connected through the overlapping relationship, as schematically illustrated in Figure 3, forming a continuous region with no false pairs. The remaining true pairs seem to be inseparable from the false pairs. Our preliminary data also suggested the overall shape of this region seems to be quite complex.

We are continuing our investigation to gain a better understanding of it. We think, after all, that a significant portion of true pairs is separable from the false ones in this parameter space, though actually separating them could turn out to be highly challenging due to the very complex shape of the region.

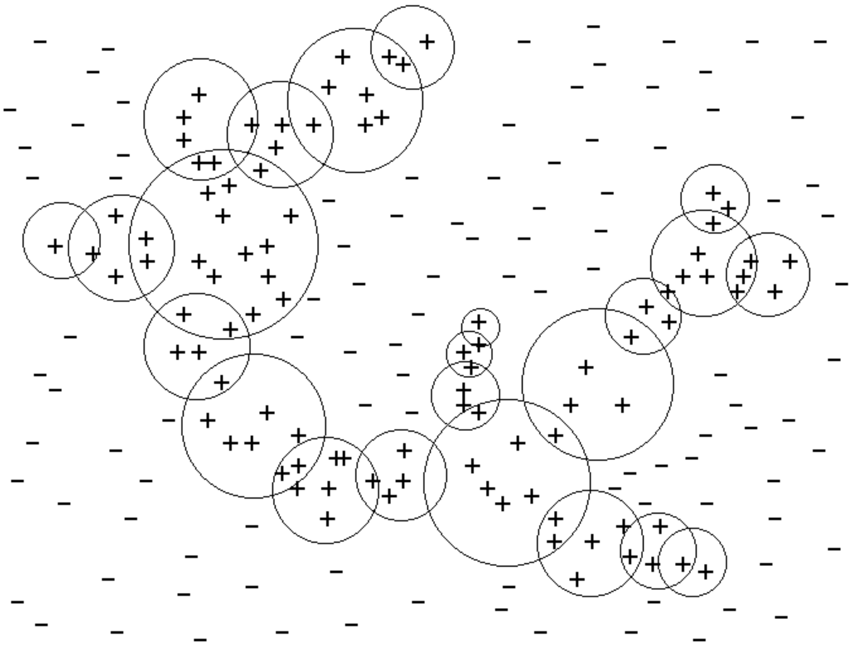


Figure 3. Schematic 2D view of the parameter space. A "+" sign represents a true pair, while a "-" sign denotes a false pair. The spheres show the regions which contain only true pairs without false ones.

### 3.2. Neural network training and results

The ultimate goal of our neural network training is to identify true sequence-structure pairs and to assess the number (or the portion) of correctly aligned residues that are structurally alignable, by our threading program. The training set consists of 50% of the 36,657 pairs that are randomly selected. Each vector consists of 10 input values as described above, and the desired output is a value $\in [0, 1]$ extracted from the second row of Table 1. We have used a commercial package, STATISTICA/neural network (StatSoft (1993)), to do the training. This software trains the neural net parameters on the training set, using the standard back-propagation algorithm for a selected number of cycles, and tests on the test set. It continues this process until the root mean square (RMS) errors for both the training and testing sets converge to similar values. During the training process, the software automatically tries different numbers of hidden nodes on each hidden layer, and keeps the network architecture with the best performance up to each time point. The total number of network architectures tried by the software depends on the time limit set by the user. Our current neural network is a result of 100 hours of training on a PC/Pentium-II. It has two hidden layers with 15 nodes on the first of them, and 12 nodes on the second. The input nodes, the first-layer hidden nodes and the second-layer hidden nodes are all fully connected.

For this particular network, we have achieved ∼11% of RMS errors on both the training and the testing sets. Figure 4 shows the detailed performance on all 36,657 pairs. What this preliminary study has achieved is that we have constructed a mapping from a raw threading score to [0, 1], and each mapped value has a well-defined meaning. For example, among threading results with a neural net score 0.6, 75% of them come from true pairs.

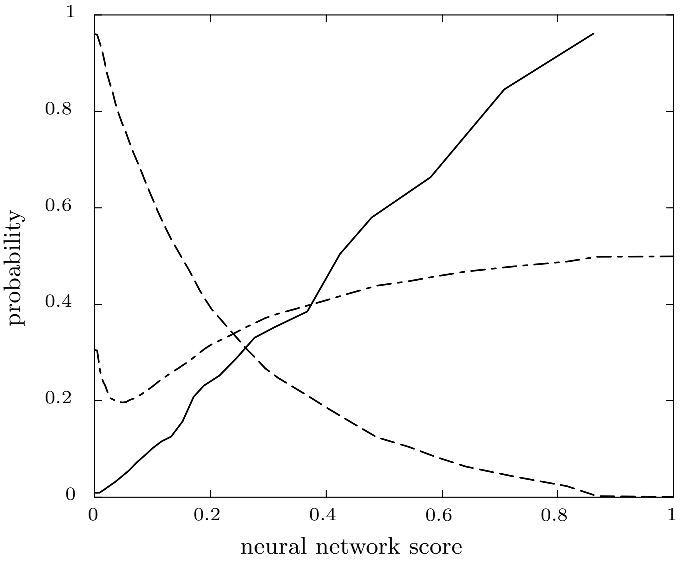


Figure 4. Prediction performance. The solid line represents the conditional probability of being a true pair for a threading given a neural network score. The dashed line represents the percentage of true pairs being above a neural network score. The dot-dashed line represents the average error of false positive and false negative if one predicts a true pair when the neural network score is larger than the threshold.

Jones has recently published a paper on using a neural network to help evaluate the significance of a threading result (Jones (1999)). The main technical difference between our work and his is that we have used "the number of structurally alignables" as the objective function for training, while he has used 0/1 (true/false) as the desired training output. In addition, we expect our investigation on the overall shape of the region occupied by the true pairs in the parameter space to lead to significant new insights into this challenging problem.

In the following, we provide a simple explanation of why the use of our objective function should give at least as good result as that of Jones. Let $\theta$ be the input vector of the neural network, and $\tau(\theta, \omega)$ be the neural network output function to match the desired output, where $\omega$ represents the network weights obtained from training. The input vector $\theta$ is generally insufficient to determine whether a threading result indicates a true pair or not. Therefore, we assume a probability of a true pair as a function of $\theta$, say $f(\theta)$. The goal of the neural network training is to minimize the error between the desired output $d$ and $\tau(\theta, \omega)$, i.e.,

$$\min_{\omega} \int L(\tau(\theta, \omega) - d) \, \mathrm{d}P(\theta), \tag{8}$$

where $P(\theta)$ represents the distribution of $\theta$ and $L > 0$ is a penalty function for the error. A typical functional form is $L(x) = x^2$. Here we do not use a special functional form, but rather assume $L$ is a convex function. When using 0/1 as desired outputs, the function in (8) has two terms, representing true pairs and false pairs. This can be written as

$$f(\theta) \, L(\tau(\theta, \omega) - 1) \; + [\, 1 \; - \; f(\theta) \,] \, L(\tau(\theta, \omega) - 0). \tag{9}$$

Since $L$ is convex function, $\lambda L(x) + (1 - \lambda) L(y) \geq L(\lambda x + (1 - \lambda) y)$. Hence, for each $\theta$, the value of (9) is larger than

$$L \; ( \; f(\theta) \, [\, \tau(\theta, \omega) - 1 \,] \; + [\, 1 \; - \; f(\theta) \,] \, \tau(\theta, \omega) \; ) = \; L \; ( \, \tau(\theta, \omega) - f(\theta) \, ) \, . \tag{10}$$

The right side of equation (10) is what our neural network minimizes, i.e., instead of minimizing the difference between the neural network output and two discrete values of 0 and 1, we minimize the difference between the neural network output and the probability of being a true pair. We conclude that our method can achieve smaller errors between the neural network output and the desired output than that of Jones.

Though our current neural network is far from being what we wish, we have tested its performance on fold recognition on a number of query proteins with a wide range of sequence identity levels against its ten native-like folds in our template library. Here, we take one query sequence (PDB code 1bgc: 158 amino acids), and thread it against a template library with 2177 unique templates of the FSSP database (Holm and Sander (1996)). The sequence identities between 1bgc and these ten protein are different. Table 2 shows the fold recognition results on those ten native-likes structures (plus 1bgc itself). The table consists of the following information: (1) the template name, (2) the number of structurally alignable residues (#alignables) between the template and 1bgc, (3) the sequence identity between 1bgc and the template for the structurally alignable regions (seq-ident), (4) the ranking in fold recognition by the raw threading score of PROSPECT (rank-0), (5) ranking by the neural network scores (rank-neural), (6) the neural network scores, and (7) the probability of the template being a correct fold for 1bgc, respectively. Ideally, these templates should be ranked from number 1 to number 11, for fold recognition.

Table 2. Threading performance with and without neural network.

| template | #alignables | seq-ident (%) | rank-0 | rank-nn | nn-score | prob. being true |
|---|---|---|---|---|---|---|
| 1bgc | 158 | 100 | 1 | 1 | 1.00 | 1.00 |
| 1cd9c | 157 | 81 | 2 | 2 | 1.00 | 1.00 |
| 1lki | 141 | 12 | 5 | 13 | 0.20 | 0.25 |
| 1alu | 140 | 16 | 3 | 5 | 0.28 | 0.31 |
| 1huw | 134 | 10 | 6 | 4 | 0.33 | 0.37 |
| 1cnt3 | 128 | 22 | 4 | 9 | 0.23 | 0.27 |
| 1xsm | 124 | 7 | 21 | 35 | 0.14 | 0.17 |
| 1r2fa | 122 | 6 | 14 | 36 | 0.14 | 0.17 |
| 1eera | 116 | 11 | 144 | 3 | 0.34 | 0.38 |
| 6prcm | 71 | 9 | 128 | 24 | 0.17 | 0.20 |
| 6prcl | 70 | 7 | 119 | 18 | 0.18 | 0.21 |

As we can see, the neural network scores generally do much better than raw threading scores in ranking the native-like folds, particularly when the sequence identities are low – that is where we need the most help in evaluating the threading results. In addition, the neural network score directly provides a probability (or reliability) of a particular threading result (query-template pair) being a true pair. In this example, all native-like folds receive a significant probability of being true.

We have applied this neural network in our CASP4 predictions, and achieved very good results (Xu et al. (2001)). PROSPECT recognized 25 correct folds from 33 prediction targets in CASP4, the highest among all participating teams. This suggests that even though this trained neural network is far from being perfect, it is highly useful.

## 4. Discussion

Confidence assessment of a threading result is a major bottleneck in improving the applicability of threading methods. Without a capability for reliable confidence assessment, we cannot process threading results automatically as BLAST or PSI-BLAST (Altschul et al. (1997)) does, nor conduct a high-throughput fold recognition by threading on a genome-scale, though this is clearly needed as over 40 genomes have been sequenced, and their genes identified. Confidence assessment for threading is clearly a much more complicated problem than sequence comparison. It involves many more scoring terms, in particular the pairwise interaction, which are hard to fit into one probabilistic model. The distribution of threading scores, as shown in Figure 2, generally does not follow the extreme distribution observed in sequence comparisons, making it hard to adapt the theories developed for sequence comparisons to protein threading. We have taken a practical approach, rather than trying to develop a rigorous model, for assessing threading results.

Our trained neural network has clearly improved our ability to interpret a threading result and a practical way to determine to what extent it can be trusted. This method also improves the sensitivity in detecting a native-like fold, compared with fold recognition by raw scores. We have demonstrated, both theoretically and through real applications, that our method to train neural networks has much better performance than a similar method (Jones (1999)). We have applied our method in threading with PROSPECT, which has significantly improved the ability of its fold recognition, as shown in Table 2.

## Acknowledgements

This work was supported by the Office of Biological and Environmental Research, U.S. Department of Energy, under Contract DE-AC05-00OR22725, managed by UT-Battelle, LLC.

## References

Abagyan, R. A. and Batalov, S. (1997). Do aligned sequences share the same fold? *J. Mol. Biol.* **273**, 355-368.

Alexandrov, N. N. (1996). SARFing the PDB. *Protein Eng.* **9**, 727-732.

Alexandrov, N. N., Nussinov, R. and Zimmer, R. M. (1996). Fast protein fold recognition via sequence to structure alignment and contact capacity potentials. In *Biocomputing: Proceedings of the* 1996 *Pacific Symposium* (Edited by L. Hunter and T. Klein), 53-72. World Scientific Publishing Co., Singapore.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* **25**, 3389-3402.

Barker, W. C., Garavelli, J. S., McGarvey, P. B., Marzec, C. R., Orcutt, B. C., Srinivasarao, G. Y., Yeh, L. L., Ledley, R. S., Mewes, H., Pfeiffer, F., Tsugita, A. and Wu, C. (1999). The PIR-international protein sequence database. *Nucleic Acids Research* **27**, 39-42.

Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. F., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. and Tasumi, M. (1977). The protein data bank: a computer based archival file for macromolecular structures. *J. Mol. Biol.* **112**, 535-542.

Bowie, J. U., Luthy, R. and Eisenberg, D. (1991). A method to identify protein sequences that fold into a known three-dimensional structure. *Science* **253**, 164-170.

Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S. and Karplus, M. (1983). CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.* **4**, 187-217.

Bryant, S. H. and Altschul, S. F. (1995). Statistics of sequence-structure threading. *Curr. Opinion Struct. Biol.* **5**, 236-244.

CASP (1995). Protein structure prediction issue. *Proteins: Struct. Funct. Genet.* **23**, 295-462.

CASP (1997). Protein structure prediction issue. *Proteins: Struct. Funct. Genet.* **Suppl. 1. 29**, 1-230.

CASP (1999). Protein structure prediction issue. *Proteins: Struct. Funct. Genet.* **Suppl. 3. 37**, 1-237.

CASP (2001). Protein structure prediction issue. *Proteins: Struct. Funct. Genet.* **Suppl. 4. 45**, 1-199.

Dayhoff, M. O. (1978). A model of evolutionary change in proteins. *Atlas of Protein Sequences and Structure* **5**, 345-352.

Finkelstein, A. V. and Ptitsyn, O. B. (1987). Why do globular proteins fit the limited set of folding patterns? *Prog. Biophys. Mol. Biol.* **50**, 171-190.

Fischer, D., Elofsson, A., Bowie, J. U. and Eisenberg, D. (1996a). Assessing the performance of fold recognition methods by means of a comprehensive benchmark. In *Biocomputing: Proceedings of the 1996 Pacific Symposium* (Edited by L. Hunter and T. Klein), 300-318. World Scientific Publishing Co., Singapore.

Fischer, D., Rice, D., Bowie, J. U. and Eisenberg, D. (1996b). Assigning amino acid sequences to 3-dimensional protein folds. *FASEB J.* **10**, 126-136.

Flockner, H., Braxenthaler, M., Lackner, P., Jaritz, M., Ortner, M. and Sippl, M. J. (1995). Progress in fold recognition. *Proteins: Struct. Funct. Genet.* **23**, 376-386.

Gerstein, M. (1998). Patterns of protein-fold usage in eight microbial genomes: a comprehensive structural census. *Proteins: Struct. Funct. Genet.* **33**, 518-534.

Gibrat, J. F., Madej, T. and Bryant, S. H. (1996). Surprising similarities in structure comparison. *Curr. Opinion Struct. Biol.* **6**, 377-385.

Godzik, A., Skolnick, J. and Kolinski, A. (1992). A topology fingerprint approach to the inverse folding problem. *J. Mol. Biol.* **227**, 227-238.

Goldstein, R. A., Luthey-Schulten, Z. A. and Wolynes, P. G. (1992). Optimal protein-folding codes from spin-glass theory. *Proc. Natl. Acad. Sci. USA,* **89**, 4918-4922.

Gonnet, G. H., Cohen, M. A. and Benner, S. A. (1992). Exhaustive matching of the entire protein sequence database. *Science* **256**, 1443-1445.

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

Hertz, J., Krogh, A. and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation.* Addison-Wesley, New York.

Holm, L. and Sander, C. (1996). Mapping the protein universe. *Science* **273**, 595-602.

Jones, D. T. (1999). GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.* **287**, 797-815.

Jones, D. T., Taylor, W. R. and Thornton, J. M. (1992). A new approach to protein fold recognition. *Nature* **358**, 86-89.

Karlin, S. and Altschul, S. F. (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. USA* **87**, 2264-2268.

Karlin, S., Dembo, A. and Kawabata, T. (1990). Statistical composition of high-scoring segments from molecular sequences. *Ann. Statist.* **18**, 571-581.

Kraulis, P. (1991). MOLSCRIPT–a program to produce both detailed and schematic plots of protein structures. *J. Appl. Cryst.* **24**, 946-950.

Levitt, M. and Chothia, C. (1981). Structural patterns in globular proteins. *Nature* **261**, 552-558.

Li, H., Helling, R., Tang, C. and Wingreen, N. (1996). Emergence of preferred structures in a simple model of protein folding. *Science* **273**, 666-669.

Li, Z. and Scheraga, H. A. (1987). Monte carlo-minimization approach to the multiple-minima problem in protein folding. *Proc. Natl. Acad. Sci. USA* **84**, 6611-6615.

Luthy, R., Bowie, J. U. and Eisenberg, D. (1992). Assessment of protein models with three-dimensional profiles. *Nature* **356**, 83-85.

Marchler-Bauer, A. and Bryant, S. H. (1997). A measure of success in fold recognition. *Trends in Biochemical Sciences,* **22**, 236-240.

Mohanty, D., Dominy, B. N., Kolinski, A., Brooks 3rd, C. L. and Skolnick, J. (1999). Correlation between knowledge-based and detailed atomic potentials: application to the unfolding of the GCN4 leucine zipper. *Proteins: Struct. Funct. Genet.* **35**, 447-452.

National Institute of General Medical Sciences (1999). Pilot projects for the protein structure initiative (structural genomics). http://www.nih.gov/grants/guide/rfa-files/RFA-GM-99-009.html, June, RFA GM-99-009.

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

Pedersen, J. T. and Moult, J. (1997). Protein folding simulations with genetic algorithms and a detailed molecular description. *J. Mol. Biol.* **269**, 240-259.

Sippl, M. J. and Weitckus, S. (1992). Detection of native-like models for amino acid sequences of unknown three-dimensional structure in a data base of known protein conformations. *Proteins: Struct. Funct. Genet.* **13**, 258-271.

Skolnick, J. and Kolinski, A. (1991). Dynamic monte carlo simulations of a new lattice model of globular protein folding, structure and dynamics. *J. Mol. Biol.* **221**, 499-531.

Smith, T. F., Conte, L. L., Bienkowska, J., Gaitatzes, C., Rogers, R. and Lathrop, R. (1997). Current limitations to protein threading approaches. *J. Comp. Biol.* **4**, 217-225.

StatSoft (1993). *STATISTICA*. 2300 East 14th Street, Tulsa, OK 74104.

Wang, Z. X. (1998). A re-estimation for the total numbers of protein folds and superfamilies. *Protein Eng.* **11**, 621-626.

Xu, D., Crawford, O. H., Locascio, P. F. and Xu, Y. (2001). A prediction experience in casp4 using the prospect prediction program. *Proteins: Struct. Funct. Genet.* . Invited publication.

Xu, Y. and Xu, D. (2000). Protein threading using PROSPECT: Design and evaluation. *Proteins: Struct. Funct. Genet.* **40**, 343-354.

Xu, Y., Xu, D. and Uberbacher, E. C. (1998). An efficient computational method for globally optimal threading. *J. Comp. Biol.* **5**, 597-614.

Protein Informatics Group, Oak Ridge National Laboratory, 1060 Commerce Park Drive, Oak Ridge, TN 37830-6480, U.S.A.
E-mail: xyn@ornl.gov

Protein Informatics Group, Life Sciences Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6480, U.S.A.
E-mail: xud@ornl.gov

Protein Informatics Group, Oak Ridge National Laboratory, 1060 Commerce Park, Oak Ridge, TN 37830-6480, U.S.A.
E-mail: vo4@ornl.gov

(Received March 2001; accepted October 2001)