PROTEINS: Structure, Function, and Bioinformatics 57:188–197 (2004)

Profile–profile Methods Provide Improved Fold-Recognition: A Study of Different Profile–profile Alignment Methods

Tomas Ohlson, Björn Wallner, and Arne Elofsson*
Stockholm Bioinformatics Center, Stockholm University, Stockholm, Sweden

ABSTRACT **To improve the detection of related proteins, it is often useful to include evolutionary information for both the query and target proteins. One method to include this information is by the use of profile–profile alignments, where a profile from the query protein is compared with the profiles from the target proteins. Profile–profile alignments can be implemented in several fundamentally different ways. The similarity between two positions can be calculated using a dot-product, a probabilistic model, or an information theoretical measure. Here, we present a large-scale comparison of different profile–profile alignment methods. We show that the profile–profile methods perform at least 30% better than standard sequence-profile methods both in their ability to recognize superfamily-related proteins and in the quality of the obtained alignments. Although the performance of all methods is quite similar, profile–profile methods that use a probabilistic scoring function have an advantage as they can create good alignments and show a good fold recognition capacity using the same gap-penalties, while the other methods need to use different parameters to obtain comparable performances. Proteins 2004; 57:188–197.** © **2004 Wiley-Liss, Inc.**

Key words: fold recognition; profile–profile alignment; PSI-BLAST; homology detection; sequence alignments

INTRODUCTION

The reliable detection of homologous protein domains is one of the oldest challenges in bioinformatics. The knowledge that two or more proteins are homologous is the most commonly used method to transfer functional knowledge from one protein to another, and it can be used for evolutionary studies and prediction of protein structure. Besides the importance to detect homologous proteins, many bioinformatical methods and tools are based on these relationships. For instance, the reliable detection of related proteins might actually improve secondary structure[1] and other prediction methods.

It has been demonstrated that methods using multiple sequences, i.e., evolutionary information, are superior to methods that only use single sequences[2] and more recently that methods that use evolutionary information for both the query and target sequences are even more efficient[3] when it comes to detecting homologous proteins. In principle, this information can be used in at least three different ways: by using, profile–profile alignments, sequence linking, and combined profile–sequence and sequence–profile searches. Profile–profile methods might be the most promising of these methods as the other methods could, at least in theory, be improved further by the use of profile–profile alignments instead of sequence–profile alignments in the different steps. However, the other methods might have computational advantages.[3]

Profile–profile alignments can be implemented in several different ways.[4–10] Although some studies have been performed comparing the performance of different profile–profile methods,[6,11] we still believe there is need for a more detailed study. The fundamental difference between different profile–profile alignment methods lies in how they calculate the score between two profile positions. A profile is a set of vectors, where each vector contains the frequency of each type of amino acid in a particular position of the multiple sequence alignment. In sequence–profile alignments, the score is calculated by extracting (the log of) the probability for an amino acid in this vector. However, in profile–profile alignments, we have to compare two frequency vectors and this can be done in several different ways, including calculating the sum of pairs, the dot-product, or a correlation coefficient between the two vectors. In addition, information about the background frequency can be used. In this study, we compare four different scoring methods: dot-product scoring (DP) as introduced in the FFAS method by Rychlewski et al.,[5] the log_aver method by von Öhsen and colleagues,[6,12] a scoring method (prob_score) based on PICASSO3 developed by Heger and Holm[13] and modified by Mittelman et al.,[11] and an information theoretical measure (prof_sim) by Yona and Levitt.[7] The different methods use different types of

Abbreviations: SCOP, the Structural Classification of Proteins database; Family, protein domains that are closely related having a common origin according to SCOP; Superfamily, protein domains of probable common origin according to SCOP; Fold, protein domains that have major structural similarities according to SCOP.

Grant sponsor: Swedish Foundation for Strategic Research; Grant sponsor: Swedish Research Council.

*Correspondence to: Arne Elofsson, Stockholm Bioinformatics Center, Stockholm University, SE-106 91 Stockholm, Sweden. E-mail: arne@sbc.su.se

Received 24 October 2003; Accepted 16 March 2004

Published online 14 May 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/prot.20184

© 2004 WILEY-LISS, INC.

comparisons between the profiles: DP uses the dot-product for a pair of profile vectors, log_aver measures the probability for the profile vectors being related according to a substitution model, prob_score is a log-odds based method, while prof_sim uses a scoring system based on an information theoretic measure of difference between the two probability distributions represented by the profiles. All the profile–profile scoring methods studied here are symmetric and take the background distribution of amino acids into account in the scoring schemes. Several profile–profile methods were recently reviewed for their ability to generate short seed alignments by Mittelman et al.[11] They showed that the best performance was obtained using probabilistic scoring methods, such as prob_score and log_aver.

It should be noted that the performance of a profile–profile method also depends on factors other than the actual scoring, such as gap-penalties, alignment methodology, and E-value calculations. To obtain the best performance, it is often necessary to optimize all these factors simultaneously. However, in this study we have focused on only one aspect (the scoring) and used identical setting for all other factors. There are several reasons why we have chosen to focus on the scoring:

- By focusing on one aspect, valuable information about the fundamental principles of profile–profile alignments can be learned.
- It is quite challenging to implement a complete system correctly as several important details are often left out in the original papers and the programs are rarely available. In contrast, all methods used here are freely available and should be easily used by others.
- It could be assumed that the methods tested here would not perform very well, and we certainly do not believe that our implementations are as good as the ones in the original articles. However, we performed a more elaborate testing than earlier studies and the increase in performance compared with PSI-BLAST is quite comparable to what has been reported earlier.
- We think LiveBench,[14] CAFASP,[15] and CASP[16] are better aimed at performing studies of complete fold recognition systems, but to achieve a better understanding of the scoring it is better to perform a more elaborate test as in this study.
- Finally, we believe the scoring is the most important aspect of a profile–profile alignment method.

Below it is shown that all profile–profile methods are better at detecting distantly related proteins and provide better alignments for these proteins than sequence–profile methods. The main reason behind the improvement is most likely that the profile–profile scoring methods are better at distinguishing evolutionary related positions from non-related positions. In the following section, we will first describe and analyze the different profile–profile comparison methods and thereafter analyze the performance of these methods using well-established benchmarks for fold recognition and alignments. We show that although the scores behave very differently, the difference in performance between the different profile–profile alignment methods is quite small. However, the probabilistic scoring methods perform very well in all tests and are less sensitive to the exact choice of gap-penalties.

RESULTS

It is well known that to obtain the optimal performance in sequence–profile alignments, it is important to optimize the profiles, by including information from substitution tables, and weighing the contribution from different sequences and other tricks.[17,18] In the development of the profile–profile methods, different schemes to obtain the best profiles have been utilized. In addition, the performance of a method also depends on the optimization of other parameters such as gap-penalties. Taking all this into account, it is straightforward to compare the performance between different profile–profile implementations. Also, few profile–profile methods are easily available to the scientific community. To solve these problems, we have implemented several profile–profile alignment methods into our freely available Palign[19] program and used identical profiles, alignment methodologies, and E-value calculations to compare them. The results presented below should, therefore, only be affected by differences due to the profile–profile score and the gap-penalties. As all methods tested here are not in all details identical to the published methods, we needed to optimize the gap-penalties as described in the methods section.

To test the ability of alignment quality for the different profile–profile methods, we created structural alignments between proteins related at different levels according to SCOP. For each structurally aligned pair of residues, we calculated the different profile–profile scores and the standard sequence-profile score. The profile–profile score was calculated using the two amino acid frequency vectors obtained from a PSI-BLAST search, denoted as α and β below. In Figure 1, it can be seen that all methods show a significant difference between random profile vectors and structurally aligned profile vectors from two proteins of the same family, while the difference is much smaller between superfamily-related residues and random positions. The scores for pairs of residues aligned at fold level are almost indistinguishable from the random scores (see Table I).The second noticeable feature in Figure 1 is that the behavior of the scores differs, most methods provide distributions that resembles Gaussian distributions, while DP (and DP_{norm}) clearly have other types of distributions. In addition, to study the distributions of the scores, we have calculated the Matthews Correlation Coefficient (MCC) between structurally aligned residues and randomly chosen pairs to measure the ability of separating these, see Table I.

Dot-Product Scoring

In the dot-product (DP) method, we use the ideas from FFAS,[5] where a dot-product is used to calculate the alignment score for a pair of profile vectors. Before calculating the score between two vectors, the vectors are balanced

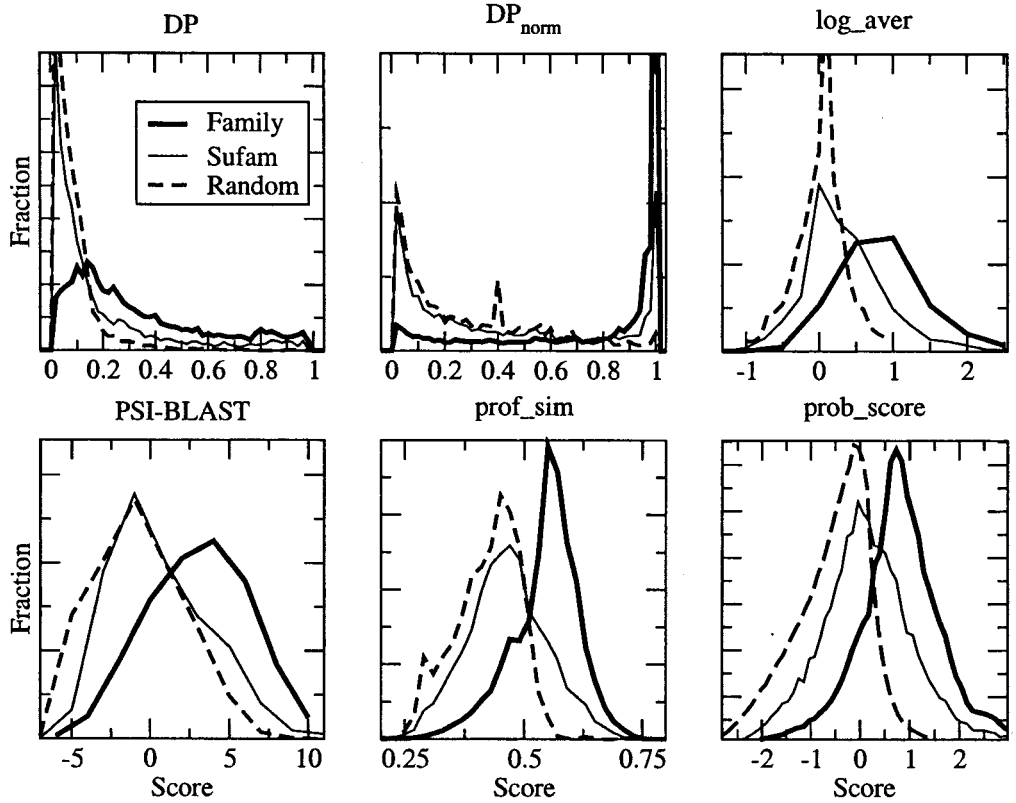


Fig. 1. Distribution of scores for the different profile–profile methods and standard profile scoring. The distribution of scores for structurally aligned residues from family-related (bold lines) and superfamily-related (thin lines) proteins are compared with scores for randomly selected residues (dashed lines). In each plot, we have plotted the fraction of residues within a certain score range against the score. The exact values of the Y-axis have been left out for clarity. The values on the X-axis represent the scores obtained before the shift values are subtracted or added.

TABLE I. Comparison of the Scores Between Structurally Aligned Residues for Proteins of Family, Superfamily, or Fold Level Relationships and Randomly Selected Positions[†]

Method	Family	Superfamily	Fold
PSI-BLAST	0.39	0.12	0.05
DP	0.41	0.22	0.09
DP$_{norm}$	0.63	0.26	0.09
log_aver	0.60	0.30	0.15
prob_score	**0.67**	**0.38**	**0.18**
prof_sim	**0.67**	0.27	0.10

[†]All calculations are made using the 10^{-3} profiles. For each method and level of similarity, the Matthews Correlation Coefficient[28] (MCC) is calculated. The best scores are marked in bold.

in three steps to obtain the best performance; without it, the performance drops significantly. First, a new vector is created:

$$\alpha'_i = \alpha_i aafreq_i \quad i = 1, \ldots, 20$$

where α_i is amino acid i in the profile vector α, α'_i is amino acid i in the modified profile vector α', and $aafreq_i$ is the frequency of amino acid i in the database. In this step, the frequencies of the most common amino acids will be enlarged, and the frequencies of the less common amino acids will be reduced. Next, the balanced vector is multiplied by 5 and added to the original vector. The value five comes from an estimated average diversity of sequences in the profiles in the database (see Rychlewski et al.[5]).

$$\alpha''_i = 5\alpha'_i + \alpha_i \quad i = 1, \ldots, 20$$

where α''_i is the amino acid frequency of the vector α''. In the last step, the final fraction score for the vector is calculated.

$$\alpha'''_i = \frac{\alpha''^2_i}{\sum_{i=1}^{20} \alpha''^2_i}.$$

In this step, the vector of amino acid frequencies becomes a probability vector, i.e., $\sum_{i=1}^{20} \alpha'''_i = 1.0$. Now, when the profile vector is balanced, the score between two vectors can be calculated using a dot-product:

$$score(\alpha,\beta) = \sum_{i=1}^{20} \alpha'''_i \beta'''_i.$$

Finally, a *shift* value is added to the score. This means that the score between two vectors is between 0 + *shift* and 1 + *shift*. Our implementation of the DP scoring differs in two

aspects from the one used in FFAS. First, we have used the methods in PSI-BLAST to include information from pseudocounts and, second, we have not rescaled the score to a standard deviation of one.

It is not obvious how the score should be interpreted, since two identical vectors, i.e., a perfect match, do not necessarily give the highest score. If $\alpha''' = (0.6,0.4,0,\ldots)$, and $\beta''' = (1,0,\ldots)$ then $score\ (\alpha,\beta) = 0.6 + shift > \text{score}(\alpha,\alpha) = 0.52 + shift$. Thus, this method favors vectors where a few amino acids are overrepresented, i.e., conserved positions. To circumvent the problem of the scores for similar vectors, we introduced a method that normalized the vectors, so that the length of all vectors are one, i.e. $\sum_{i=1}^{20}\alpha_i'''^2 = 1.0$. We will refer to this scoring method as DP_{norm}. Now α''' and β''' becomes $\alpha'''_{norm} = (0.83,0.55,0,\ldots)$, and $\beta'''_{norm} = (1,0,\ldots)$, respectively, after normalization, which gives $score(\alpha,\beta) = 0.83 + shift < score(\alpha,\alpha) = 1 + shift$. After the normalization, the vectors no longer represent probabilities. Instead, the vectors represent orientations and the score between two vectors is now related to the angle between the vectors. So if α and β are parallel, i.e. they have the same distribution of amino acids, they will get the maximum score. DP_{norm} makes the comparison between two vectors geometrically reasonable.

In Figure 1, it can be seen that the scores from DP and DP_{norm} clearly do not follow a Gaussian distribution. In DP most scores for unrelated positions are close to the minimum value ($0 + shift$) with only a small fraction obtaining high scores. The scores for family-related residues have a broad distribution between the maximum and minimum scores and a large fraction of these are clearly separated from the random scores. The superfamily-related residues score slightly higher than the random pairs, but the scores follow a similar distribution as the random scores. In Table I, it can be seen that on all levels the DP scoring scheme is better at identifying the structurally aligned residues than standard sequence–profile methods. However, the separation (as measured by MCC) is not as good as the best profile–profile methods at the family level. At the superfamily and fold levels, the MCC values (0.22 and 0.09) are comparable with the other profile–profile methods.

The scores for DP_{norm} behave in a manner opposite to the DP scores (see Fig. 1). Most of the aligned residues have scores that are quite close to the maximum score ($1 + shift$), meaning that the aligned residues profile vectors are oriented in the same direction, while the random scores have a broad distribution between the maximum and minimum scores. The separation as measured by the MCC for family and superfamily related residues is better than for DP (Table I). The MCC for family-related pairs is 0.63 and for superfamily-related pairs 0.26. However, it was observed that DP_{norm} did not perform as well as DP in the fold recognition and alignment benchmarks analyzed below and, therefore, the results from DP_{norm} will not be included in the studies of fold recognition abilities and alignment qualities.

prof_sim

The prof_sim[7] scoring system is based on an information theoretic measure of difference between the two probability distributions represented by the profiles. They combine a divergence and a similarity score in their scoring function.

First the divergence score (Kullback-Leibler divergence, KL) for two profile vectors is defined to be:

$$D^{KL}(\gamma,\delta) = \sum_{k=1}^{20}\gamma_k \log_2\frac{\gamma_k}{\delta_k}$$

and then the Jensen-Shannon divergence is defined as

$$D_\lambda^{JS}(\alpha,\beta) = \lambda D^{KL}(\alpha,r) + (1-\lambda)D^{KL}(\beta,r)$$

where $r = \lambda\alpha + (1 - \lambda)\beta$. In this investigation, λ was set to 1/2. Finally the significance score is defined as:

$$S = D^{JS}(r,\ aafreq)$$

where *aafreq* is the frequency of amino acids in the database. Now the prof_sim score is calculated by:

$$score(\alpha,\beta) = \frac{1}{2}(1 - D^{JS}(\alpha,\beta))(1 + S)$$

Also here a *shift* value is added to the score. Note that the function $D^{KL}(\alpha,\beta)$ is not symmetric, but since we are using $\lambda = 1/2$, the score is symmetric, i.e., $score(\alpha,\beta) = score(\beta,\alpha)$.

In prof_sim, high scores are only obtained if D^{JS} is low, i.e., α and β are similar to each other and S is high, i.e., r is different from the background distribution. The prof_sim score shows a Gaussian and overlapping distribution with a peak of a family-related hit at 1.7 standard deviations higher than for the random hits, and the average scores of superfamily-related hits are between the random and family-related scores (Fig. 1). The identification of family-related pairs is together with prob_score the best of all methods (MCC = 0.67) and for superfamily- and fold-related pairs, the separation is almost as good as for log_aver (Table I).

Probabilistic Scoring Methods

Mittelman et al.[11] showed that several probabilistic methods showed a good performance. Here, we have studied two probabilistic scoring methods, log_aver[6,12] and prob_score, which is based on the Picasso method developed by Heger and Holm.[13]

log_aver

The log_aver[6,12] method is the only method that does not use the exact profiles from PSI-BLAST, but the distribution frequencies directly. The reason is that the substitution matrix is included in the comparison. The log_aver score is defined as:

$$score(\alpha,\beta) = \ln\sum_{i=1}^{20}\sum_{j=1}^{20}\alpha_i\beta_j \cdot \exp((\ln 2/2)\cdot BLOSUM62_{i,j})\,.$$

where $BLOSUM62_{i,j}$ is the value in the BLOSUM62 substitution matrix for amino acid i replaced with j. Finally, a *shift* value is added to the score. However, during optimization it was found that the best *shift* value was zero. The score is basically a geometric mean of the BLOSUM62 score for all amino acids in a position of the multiple sequence alignment from one profile compared with all amino acids from the other, or in other words the sum of the probability to replace all amino acids found in the α profile with the ones found in the β profile. Since this method includes a substitution matrix in the scoring scheme, the highest possible score is obtained if the vectors α and β have a similar distribution and are conserved. The lowest score is obtained if the distribution between α and β contains dissimilar amino acids. For log_aver, the distribution of scores is found within an interval of $(-1, 2)$ (Fig. 1). All groups of scores have Gaussian-like distributions, where the random alignments peak close to zero and the family-related alignments peak about two standard deviations higher. The superfamily-related scores show an intermediate behavior. It can be seen that on superfamily and fold levels, log_aver show a better performance in separation between structurally aligned residues and random ones than DP and prof_sim with MCC values of 0.30 and 0.15, while on the family level the performance was not quite as good as for prof_sim (Table I).

prob_score

The second probabilistic scoring method in this study was originally introduced by Heger and Holm[13] for the comparison of protein families represented by profiles. The original PICASSO scoring method is defined by:

$$score(\alpha,\beta) = \sum_{i=1}^{20} \alpha_i \log \frac{\beta_i}{aafreq_i}.$$

Mittelman et al.[11] modified the PICASSO scoring scheme into a symmetric scoring scheme referred to as PICASSO3:

$$score(\alpha,\beta) = \sum_{i=1}^{20} \alpha_i \log \frac{\beta_i}{aafreq_i} + \sum_{i=1}^{20} \beta_i \log \frac{\alpha_i}{aafreq_i}$$

Finally, a *shift* value is added to the score. We have used this probabilistic scoring method (prob_score) without the modification to the profiles used in PICASSO3. Our implementation differs slightly from the one used by Mittelman et al.[11] For instance, we have included substitution matrices and not the raw frequencies, as this provided a better performance on the fold recognition test (data not shown). In prob_score, the highest score is obtained when two similar conserved positions are aligned with each other, while positions that have a similar amino acid frequency as the background distribution will have intermediate scores. On all levels of relationship, the MCC value for prob_score is superior to all the other methods.

The two probabilistic methods, log_aver and prob_score, performed very well at separating the structurally aligned residues from random residues (Table I). On the superfamily and fold level, prob_score is the best method, followed by log_aver. Particularly impressive is that these methods show an MCC of 0.15 and 0.18 for aligned positions at the fold level, i.e., for residues that should not be evolutionary related but only structurally related.

Fold Recognition

From the results described above, one could assume that prob_score should be better than the other methods at fold recognition, as it provides the best identification of distantly related residues. Certainly, other factors could influence the fold recognition performance. To minimize the dependency of these factors, we have optimized the gap- and shift-penalties for all methods (see below). We have compared the performance of the different profile–profile methods with PSI-BLAST using a similar benchmarking strategy as in several earlier studies.[2,3,20] To highlight differences in performances, we choose to not plot sensitivity vs. specificity directly against each other. Instead, we have plotted sensitivity vs. $log(1 - specificity)$, i.e., the log of the error rate, as in a ROC plot.[21] We have used two different E-value cutoffs (10^{-2} and 10^{-3}) to create the profiles as these have been shown to provide the best performance in earlier studies.[3] In addition to this plot, we have, as in our earlier study,[3] chosen to specifically study the performance at 10% error rate for superfamily-related proteins and at 1% error rate for family-related proteins.

In Figure 2 and in Table II, it can be seen that for low error rates (<1%) log_aver, prob_score, and DP clearly perform better than prof_sim and PSI-BLAST detecting more than 75% of the family-related proteins at 1% error rate. The prof_sim method performs similar to PSI-BLAST at the 1% error rate, detecting up to 69% of the proteins compared with 67% for PSI-BLAST. However, as can be seen in Figure 2, all profile–profile methods do perform better than PSI-BLAST at higher error rates. At the superfamily level, all profile–profile methods perform better than PSI-BLAST. Just as for the family level, it can be seen that the performance of log_aver, DP, and prob_score are superior to prof_sim at low error rates (see Fig. 2). At 10% error rate, the profile–profile methods detect about 18–23% of the proteins using the 10^{-3} profiles and 20–25% using the 10^{-2} profiles. In comparison PSI-BLAST only detects 12 and 15% respectively, i.e., the profile–profile methods detect more than 30% additional superfamily-related proteins. At lower error rates, the performance of prob_score is superior to the other methods.

For all methods, the 10^{-2} profiles perform better than the 10^{-3} profiles (Table II). However, the ability to detect superfamily-related proteins drops significantly at around 1% error rate using the profiles for most methods compared with the profiles (data not shown). A similar drop can be seen at around 0.5% error rate at the family level using the 10^{-2} profiles. The increased sensitivity using a looser E-value cutoff is similar to what has been reported earlier for PSI-BLAST.[3] However, the profile–profile methods seem to be slightly better than PSI-BLAST at maintaining high specificities using the 10^{-2} profile (Fig. 2). It

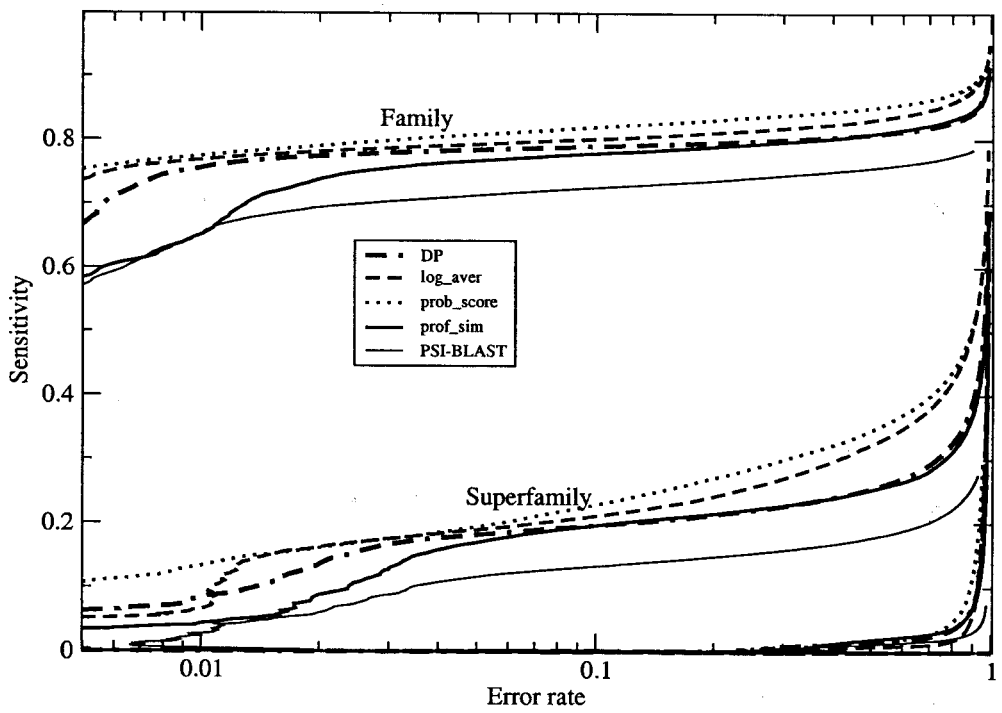


Fig. 2. ROC plot for profile–profile methods and PSI-BLAST using 10^{-2} profiles. For each score, the sensitivity is plotted against the log of the error rate. The top curves represent family level performance, the middle set of curves represent superfamily performance and the bottom curves represent fold performance.

TABLE II. Sensitivity of all Profile–Profile Methods Using Either Profiles Made with an E-Value Cutoff of 10^{-2} or 10^{-3} and PSI-BLAST†

Method	E-value	Family sensitivity (%)	Superfamily sensitivity (%)
PSI-BLAST	10^{-2}	67	15
PSI-BLAST	10^{-3}	66	12
DP	10^{-2}	76	20
DP	10^{-3}	74	16
log_aver	10^{-2}	77	21
log_aver	10^{-3}	76	18
prob_score	10^{-2}	**78**	**25**
prob_score	10^{-3}	**77**	**23**
prof_sim	10^{-2}	65	20
prof_sim	10^{-3}	69	17

†The sensitivities at 1% (family) and 10% (superfamily) error rates are shown. The best performing methods are marked in bold.

TABLE III. Profile–Profile Alignment Quality, as Measured by Average MaxSub Score, for the Profile–Profile Methods and for Standard Sequence–Profile Alignments (PSI-BLAST)†

Method	Family	Superfamily
PSI-BLAST	0.53	0.12
DP	0.56	0.17
log_aver	0.55	**0.18**
prob_score	0.57	0.17
prof_sim	**0.58**	**0.18**

†The comparisons were done using 841 pairs related on the family level and 1,039 pairs on the superfamily level using local alignments. All studies were performed using the 10^{-3} profiles. The estimated error of the measured alignment qualities is in the order of 0.015. The best performing methods are marked in bold.

should also be noted that the performance below 1% error rate is quite sensitive to the exact choice of gap-penalties and, therefore, we trust more in the increased performance at 1% and higher error rates. However, it is not unlikely that prob_score has an advantage at low error rates compared to the other methods.

Alignment Quality

In addition to studying the ability to recognize proteins at different levels, we have studied the quality of alignments. This was performed by using two sets of protein pairs, one set of 841 protein pairs that belong to the same family according to SCOP, and one set of 1,039 pairs that belong to the same superfamily but different families. The quality of the alignments was measured by MaxSub,[22] a measure that should be one for a perfect model and zero for a completely wrong model. For each method, we optimized the gap- and shift-penalties and the average MaxSub score using the optimized penalties for the two test sets (shown in Table III).

From Table III, it is clear that all methods provide better alignments than PSI-BLAST. For family-related pairs, the improvement is quite small from an average MaxSub score of 0.53 to average scores of 0.55 to 0.58, while the increase is larger for superfamily-related pairs, from 0.12 to 0.17-0.18, i.e., an increase of 40–50%. The alignment quality differences between the different profile–profile method

are within the error margin, but it is possible that prof_sim provides slightly better alignments than the other methods.

DISCUSSION

What Type of Information Should Be Used in a Profile–Profile Score

When calculating a score from two profile vectors, there are two factors that have to be taken into account. First, a high score should be obtained only if the two vectors are similar to each other. Second, even if the two vectors are similar to each other but they correspond to a random distribution, they should not give high scores. In addition, it is important to use prior information in the form of a substitution matrix to obtain the best possible performance. These problems have been solved differently by the different profile–profile methods.

The difference between DP and DP_{norm} lies in the normalization of the vectors. In DP, the vectors are normalized to a sum of one, while in DP_{norm} they are normalized to a length of one. It was clear in our studies that DP was superior to DP_{norm} (data not shown). The reason is most likely that higher scores are given to profile vectors that are similar to the background distribution of amino acids in DP_{norm}. This shows that DP in an indirect way: by shortening the length of the vectors that have a broad distribution of amino acids lowers the scores that are given to less conserved residues.

Since log_aver includes a substitution matrix in the scoring scheme, the highest possible score is obtained if the vectors α and β have the same distribution and at the same time are over-represented by amino acids that give high scores in the BLOSUM matrix. Profile vectors that represent a random distribution of amino acids will have intermediate scores in log_aver, just as two randomly chosen amino acids in a sequence alignment.

The scoring function in prob_score behaves quite similarly to log_aver as the amino-acid frequencies that are similar to the background frequency get intermediate scores, while over- and under-estimated scores obtain high and low scores, respectively. The major difference between prob_score and log_aver is that prob_score uses the substitution matrix information in the creation of the profile vectors while log_aver uses the BLOSUM matrix directly.

The prof_sim score is based on a divergence measure between two vectors and a significance score that measures how different the vectors α and β are from the background distribution. To obtain a high score, it is necessary to have a low divergence and a high significance. The combination of two measures to achieve the two demanded effects seems more complicated than in log_aver and prob_score.

The behavior of the DP scores surprised us. The distribution of the scores in DP are very different from all other methods. The dot-product mainly provides two distinct distributions (Fig. 1). It can be seen that only a small fraction of the superfamily-related residues actually have significantly higher scores than random alignments for DP. This type of distribution might create instabilities in the alignments, i.e., aligning some parts with very good scores but completely missing other parts as these scores are indistinguishable from random scores. Instinctively, it seems as if the DP scores should be non-optimal, but it is possible that giving very high scores to a few (more conserved) residues actually ignores noise. It is possible that this scoring scheme is able to detect strong signals in a short region of the protein, and in earlier studies DP-based methods have shown a very good performance.[14,23] Our results indicate that DP perform as well as the best profile–profile alignment methods for fold recognition, and provide quite good alignments. However, it was also noted that to obtain the best alignments with DP, it was necessary to use significantly lower gap-penalties than in fold recognition, probably due to the fact that only the most similar regions were aligned otherwise (see Table IV).

TABLE IV. Optimized Gap-Open (GO), Gap-Extension (GE), and Shift Parameters for All of the Methods[†]

	Fold recognition			Alignment		
Method	GO	GE	Shift	GO	GE	Shift
PSI-BLAST	10.0	1.00	—	5.5	0.75	—
DP	0.71	0.30	−0.08	0.07	0.02	−0.05
log_aver	1.80	0.03	0.00	1.00	0.03	0.00
prob_score	1.35	0.30	+0.50	1.35	0.30	+0.50
prof_sim	1.00	0.10	−0.45	0.30	0.01	−0.38

[†]The left three columns correspond to the optimal values for fold recognition and the three right columns correspond to the alignments. It should be noted that no shift value was optimized for PSI-BLAST.

All Profile–Profile Methods Perform Better Than PSI-BLAST

We have used three different approaches to compare different profile–profile methods. First, we analyzed the ability to separate structurally aligned residues from random ones in an attempt to study how well the different scoring methods detect related residues. The main advantage of this analysis is that it is completely independent of gap-penalties. Second, we studied the ability to detect related proteins in a fold-recognition benchmark. Finally, we studied the quality of the alignments. The correlation between the results from the different tests is not perfect. For instance, prof_sim performed quite well in all but the fold recognition benchmarks, while DP did not perform very well in the first test. This highlights the fact that fold recognition and alignments of distantly related proteins is a complex problem and that many factors might influence the results.

However, what is clear is that all the methods are significantly better at fold-recognition than standard sequence–profile methods, which is in agreement with earlier studies.[5–7,9,10] The alignment qualities of the profile–profile methods are also better than for standard sequence–profile alignments, which is in agreement with Sadreyev and Grishin[8] who showed that profile–profile methods created better initial alignments than

TABLE V. Fold Recognition and Alignment Quality Performance at Superfamily Level Using Gap- and Shift-Parameters Either Optimized for Fold Recognition (FR-gap) or Alignment (AL-gap)[†]

	Fold recognition (%)		Alignment quality	
Method	FR-gap	AL-gap	FR-gap	AL-gap
PSI-BLAST	12	11	0.09	0.12
DP	16	9	0.04	0.17
log_aver	18	16	**0.17**	**0.18**
prob_score	**23**	**23**	**0.17**	0.17
prof_sim	17	13	0.07	**0.18**

[†]The fold recognition is measured as the sensitivity obtained at 10% error rates for superfamily-related proteins and the alignment quality is measured as the average MaxSub score for 1,039 superfamily-related pairs. All studies are performed using the 10^{-3} profiles. It should be noted that the optimal parameters for prob_score are identical in the alignment and fold recognition tests and, therefore, the scores are identical. The best performing methods are marked in bold.

PSI-BLAST and Jaroszewski et al.[24] who showed that a modified version of FFAS, called FFAS-R, made better alignments than PSI-BLAST.

Although our implementation of the different profile–profile methods is not identical to the ones in the original articles, we see a similar increase in performance. At the superfamily level, we obtained an increased detection rate of 30% or more depending on the profiles being used and a similar increase in alignment quality was also observed. This shows that in the near future to perform sensitive searches and obtain optimal alignments, profile–profile methods should become the standard tool, if the computational cost can be justified.

Stability of Gap-Penalties

To obtain the best performance, it was necessary to optimize the gap- and shift-parameters for fold recognition and alignment qualities independently. As can be seen in Table IV, the same set of parameters could be used for both alignment and fold recognition only for prob_score. The most significant changes in the parameters were the significantly lower gap-opening penalty for DP, and, to a lesser degree, for prof_sim, needed to obtain the best alignments. However, several of the parameters for the other methods also had to be changed. The lowering of the gap-penalties or the use of a higher shift value will result in longer alignments, possibly at a price of lower specificity. During the manual optimization it was also observed that some methods seemed to be more sensitive to the exact choice of parameters than other methods. However, due to the large amount of computer time needed to perform the optimization, it is not trivial to get an accurate measure of the parameter stability.

To try to examine the dependency of the gap- and shift-parameters, we used the parameters optimized for fold recognition on the alignment benchmark and vice versa and studied the performance on the superfamily level (Table V).With the parameters optimized for alignment qualities, the alignment performance is quite similar for all profile–profile methods, with average MaxSub scores of 0.17 to 0.18, but if the fold recognition parameters were used, only prob_score (which uses the identical parameters) and log_aver (0.17) maintained an acceptable alignment performance. The alignment qualities dropped significantly for DP and prof_sim, to average MaxSub scores of 0.04 and 0.07. This is actually not better than for PSI-BLAST, which obtains an average MaxSub score of 0.09 using the fold recognition parameters. A similar result is obtained if the alignment parameters are used for fold recognition (see left side of Table V). Again, only prob_score (23%) and to a lesser extent log_aver (16%) performed acceptably using the parameters optimized for alignment quality. The superfamily sensitivity dropped to 9% and 13% for DP and prof_sim, respectively.

From the results in Table V, it is clear that the method that needs the largest lowering of gap-penalties (DP) also loses the most in performance. The prof_sim method, which also needs a significantly lower gap-penalty and increased shift value, also performs worse when the non-optimal parameters are used. The non-Gaussian distribution of the scores in DP might focus the alignments used in fold-recognition to the most conserved regions and, therefore, it seems to be necessary to use different gap-penalties to align a longer region of the proteins. However, we have no good explanation why it is necessary to use different gap-penalties for prof_sim. It should also be highlighted that prob_score was the only method that could use exactly the same gap-penalties and still perform as well as the best methods both for alignments and fold recognition.

Conclusions

In this study, we have shown that several different profile–profile methods perform significantly better than standard sequence-profile methods. The profile–profile methods show a greater ability to identify structurally related residues, and provide better recognition and better alignments than standard sequence-profile methods. The different profile–profile methods perform quite similarly if the gap-penalties are optimized individually for alignment and fold recognition abilities. However, it seems as if the probabilistic scoring functions (log_aver and prob_score) has a slight advantage as these are the only methods that show good performance in both fold recognition and alignment quality using identical parameters. It is also possible that prob_score is superior to the other methods for fold recognition at high specificities.

The distribution of the scores in DP appears not to be ideal as it is necessary to use quite different gap-penalties to obtain high-quality alignments in comparison with the optimal penalties for fold recognition and vice versa. The good performance of log_aver and prob_score is in agreement with the results from Mittelman et al.[11] who showed that probabilistic scoring functions yield more accurate short seed alignments. The probabilistic scoring functions includes in a natural way the two most important factors of a profile–profile scoring function: the ability to score similar sequences high but not score two identical less conserved positions high.

This study highlights that a better understanding of how different profile–profile methods perform should be useful for improving these methods. It is not obvious what the best distribution separating related and unrelated residues should look like, even if our intuition would tell us that the Gaussian distributions are good. It is promising that the different profile–profile methods perform quite well compared with PSI-BLAST. Further, it seems as if a good ability to identify related residues is a requirement for optimal performance. However, it is also clear that an increased understanding of what factors affect the performance of fold recognition and alignment qualities has to be obtained if even better methods are developed.

METHODS

Profiles and Alignments

For all methods, we have used the profiles obtained after ten iterations of PSI-BLAST[25] version 2.2.2. We have used two different set of E-value cutoffs (10^{-3} and 10^{-2}) when creating the profiles, and all other parameters at default setting. The search was performed against nrdb90 from EBI.[26] The frequency profiles were back-calculated from the log-profiles obtained from PSI-BLAST for all methods except log_aver. The log_aver profiles were obtained directly from the amino acid frequencies in the multiple sequence alignments. The reason to not use the raw frequencies in the other methods is that they do not implicitly use a substitution model that can compensate for incomplete sampling and the raw frequencies gave significantly worse results.

Study of Profile–Profile Scores

To investigate how the different profile–profile scores behaved, we computed the scores for structurally aligned positions between two proteins related at different SCOP-levels. The structural alignments were performed using Structal.[27] These scores were compared with the scores from unrelated positions, obtained from pairs of randomly chosen positions. On the family level, up to ten proteins from each family were used and aligned with the other proteins from the same family. At the superfamily level, a similar comparison was made as on the family level, with the exception that the aligned proteins belonged to different families, and at the fold level no proteins from the same superfamily were included. For each level of similarity and profile–profile method, the scores from the structurally aligned positions were compared with the scores for random positions. The distribution of these scores were plotted in Figure 1 and, in addition, the highest Matthews Correlation Coefficients[28] separating the structurally aligned from the random positions were calculated for each method and level of similarity (see Table I) where TP is the number of structurally aligned positions identify.

Studies of Fold Recognition

All methods were implemented into the Palign[19] package that is available from http://www.sbc.su.se/~arne/palign/. Throughout this study, only local alignments were used as we showed in an earlier study that this provides good results and a comparison with PSI-BLAST is more straightforward.[3] The same benchmark set as used in our earlier study[3] was used here. This dataset contains a subset of SCOP, where no two protein domains have more than 75% sequence identity, resulting in 4,972 domains. Profiles were constructed for all the protein domains in the benchmark data set. Each of the profiles were then used to search for related profiles. In contrast to our earlier studies,[2,3] we chose to plot sensitivity, defined as the fraction of all related pairs found at a certain cutoff, versus error rate, as in a ROC plot.[29] The error rate was defined as the fraction of all pairs that are incorrect and with a better score than a cutoff.

Besides plotting sensitivity versus error rate, the methods of fold recognition performance were compared by measuring their sensitivity at fixed error rates.[3] At the family level, the sensitivity was measured at 1% error rate (family sensitivity) and at the superfamily level at 10% error rate (superfamily sensitivity). The sensitivity measures the fraction of related members in a fold/superfamily/family that the method can detect. The ROC-plots were calculated using E-values calculated in the same way as in FASTA[30] and as described earlier.

Studies of Alignment Quality

Alignment quality was studied using a set of proteins obtained from the benchmark set. To avoid bias, only a maximum of ten members from each family was used. This resulted in 841 family related pairs that were studied for alignment quality. We also studied the alignment quality of 1,039 pairs that were related on the superfamily level, using at most one member from each family. For each method the gap opening, gap extension, and shift values were optimized (see below). In all cases, local alignments were used as we could not detect any major improvement using global alignments (data not shown). The PSI-BLAST alignments were produces using standard sequence–profile alignments with the palign program.

There are many fundamentally different methods to study alignment quality.[31] Here, we chose to use a similar approach as in LiveBench,[23] CASP,[32] and CAFASP.[33] For each alignment, we created a model of the query protein and compared the structure of this model with the correct structure. We used MaxSub,[22] which finds the largest subset of atoms of a model that superimposes well over the experimental model. The results obtained using other methods, such as LGscore,[31] were almost identical and therefore we only report the average MaxSub score here.

Optimization of Gap and Shift Parameters

To optimize the gap-penalties and the shift value for fold recognition, we used a smaller test-set (256 proteins) and searched the database of 4,972 proteins. For each method, we performed a grid-search starting with five values for gap-opening, gap-extension, and shift values. The parameter search was then extended and tuned toward the direction that produced the best results until a stable maximum was found. The parameters that produced the best sensitivity on the family or superfamily level were

then tested on the larger test set, and in the analysis above we have used the set of parameters that performed best at this set (Table IV). In total, hundreds of parameters were tested on the small set for each method and at least five different parameters were tested on the large set. In a few cases, a few percentages of higher sensitivity on the family level could be obtained at the price of a slightly decreased performance on the superfamily level or vice versa. In these cases, we chose the set of parameters that we thought provided a good balance between the family and superfamily performance.

The gap- and shift-parameters were independently optimized using the full alignment benchmark. The optimization was started using an identical grid search as for the fold recognition, but it was noted that for some of the methods lower gap penalties and/or different shift parameters had to be used to obtain the best possible alignments and, therefore, the searches were extended. The set of parameters that produced the best alignments are shown in Table IV.

ACKNOWLEDGMENTS

This work was supported by grants from the Swedish Foundation for Strategic Research and the Swedish Research Council to A.E. We are grateful for help with the structural alignments from Erik Sandelin. Håkan Viklund, Bob MacCallum, and other scientists at SBC provided interesting discussions and ideas. We are also thankful to one of the referees who pointed out the differences between our implementations and the original ones.

REFERENCES

1. Rost B, Sander C. Prediction of protein secondary structure at better than 70% accuracy. J Mol Biol 1993;232:584–599.
2. Lindahl E, Elofsson A. Identification of related proteins on family, superfamily and fold level. J Mol Biol 2000;295:613–625.
3. Wallner B, Fang H, Ohlson T, Frey-Skött J, Elofsson A. Using evolutionary information for the query and target improves fold recognition. Proteins 2004;54:342–350.
4. Fischer, D. Hybrid fold recognition: combining sequence derived properties with evolutionary information. In Altman RB, Dunker AK, Hunter L, Klien TE, editors. Pacific Symposium on Biocomputing, vol. 5. River Edge, NJ: World Scientific; 2000. p 116–127.
5. 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.
6. von Öhsen N, Sommer I, Zimmer R. Profile–profile alignments: a powerful tool for protein structure prediction. In: Altman RB, Dunker AK, Hunter L, Jung TA, Klein TE., eds. Pacific Symposium on Biocomputing. River Edge, NJ: World Scientific; 2003. p 252-263.
7. Yona G, Levitt M. Within the twilight zone: a sensitive profile–profile comparison tool based on information theory. J Mol Biol 2002;315:1257–1275.
8. Sadreyev R, Grishin N. COMPASS: A Tool for comparison of multiple protein alignments with assessment of statistical significance. J Mol Biol 2003;326:317–336.
9. Edgar R, Sjolander K. SATCHMO: sequence alignment and tree construction using hidden Markov models. Bioinformatics 2003;19: 1404–1411.
10. Pei J, Sadreyev R, Grishin NV. PCMA: fast and accurate multiple sequence alignment based on profile consistency. Bioinformatics 2003;19:427–428.
11. Mittelman D, Sadreyev R, Grishin N. Probabilistic scoring measures for profile–profile comparison yield more accurate short seed alignments. Bioinformatics 2003;19:1531–1539.
12. von Öhsen N, Zimmer R. Improving profile–profile alignments via log average scoring. In Workshop on Algorithmic Bioinformatics, 2001;11–26.
13. Heger A, Holm L. Exhaustive enumeration of protein domain families. J Mol Biol 2003;328:749–767..
14. Rychlewski L, Fischer D, Elofsson A. LiveBench-6: large-scale automated evaluation of protein structure prediction servers. Proteins 2003;53:542–547.
15. Fischer D, Rychlewski L, Dunbrack RL, Ortiz AR, Elofsson A. CAFASP3: The third critical assessment of Fully Automated protein structure prediction methods. Proteins 2003;53:503–516.
16. Moult J, Hubbard T, Bryant SH, Fidelis K, Pedersen JT. Critical assessment of methods of proteins structure predictions (CASP): Round II. Proteins (Suppl) 1997;1:2–6.
17. Luthy R, Xenarios I, Bucher P. Improving the sensitivity of the sequence profile method. Protein Sci 1994;1:139–146.
18. Schaffer A, Aravind L, Madden T, Shavirin S, Spouge J, Wolf Y, Koonin E, Altschul S. Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. Nucleic Acids Res 2001;29:2994–3005.
19. Elofsson A. A study on how to best align protein sequences. Proteins 2002;15:330–339.
20. Hargbo J, Elofsson A. A study of hidden Markov models that use predicted secondary structures for fold recognition. Proteins 1999; 36:68–87.
21. Park J, Karplus K, Barrett C, Hughey R, Haussler D, Hubbard T, Chotia, C. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. J Mol Biol 1998;284:1201–1210.
22. Siew N, Elofsson A, Rychlewski L, Fischer D. MaxSub: An automated measure to assess the quality of protein structure predictions. Bionformatics 2000;16:776–785.
23. Bujnicki J, Elofsson A, Fischer D, Rychlewski L. Livebench-2: large-scale automated evaluation of protein structure prediction servers. Proteins 45 (Suppl) 2001;5:184–191.
24. Jaroszewski L, Rychlewski L, Godzik A. Improving the quality of twilight-zone alignments. Protein Sci 2000;9:1487–1496.
25. Altschul S, Madden T, Schaffer A, Zhang J, Zhang Z, Miller W, Lipman D. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 1997;25: 3389–3402.
26. Holm L, Sander C. Removing near-neighbour redundancy from large protein sequence collections. Bioinformatics 1998;14:423–429.
27. Gerstein, M. and Levitt, M. Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. Protein Sci 1998;7:445–456.
28. Matthews, B. Comparison of predicted and observed secondary structure, of T4 phage lysozyme. Biochim Biophys Acta 1996;405: 442–451.
29. Campbell G. Advances in statistical methodology for the evaluation of diagnostic and laboratory tests. Stat Med 1994;13:499–508.
30. Pearson WR, Lipman DJ. Improved tools for biological sequence analysis. Proc Natl Acad Sci USA 1988;85:2444–2448.
31. Cristobal S, Zemla A, Fischer D, Rychlewski L, Elofsson A. A study of quality measures for protein threading models. BMC Bioinformatics 2001;2:5.
32. Moult J, Hubbard T, Fidelis K, Pedersen J. Critical assessment of methods of protein structure predictions (CASP): Round III. Proteins (Suppl) 1999;3:2–6.
33. Fischer D, Elofsson A, Rychlewski L, Pazos F, Valencia A, Rost B, Ortiz A, Dunbrack R. CAFASP2: The critical assessment of fully automated structure prediction methods. Proteins (Suppl) 2001;5: 171–183.