Protein Sequence Threading, the Alignment Problem, and a Two-Step Strategy

THOMAS HUBER,[1] ANDREW E. TORDA[2]

[1]ANU Supercomputing Facility and [2]Research School of Chemistry, Australian National University, Canberra ACT 0200, Australia

Received 18 September 1998; accepted 2 May 1999

ABSTRACT: Conventionally, protein structure prediction via "threading" relies on some nonoptimal method to align a protein sequence to each member of a library of known structures. We show how a score function (force field) can be modified so as to allow the direct application of a dynamic programming algorithm to the problem. This involves an approximation whose damage can be minimized by an optimization process during score function parameter determination. The method is compared to sequence to structure alignments using a more conventional pair-wise score function and the frozen approximation. The new method produces results comparable to the frozen approximation, but is faster and has fewer adjustable parameters. It is also free of memory of the template's original amino acid sequence, and does not suffer from a problem of nonconvergence, which can be shown to occur with the frozen approximation. Alignments generated by the simplified score function can then be ranked using a second score function with the approximations removed. © 1999 John Wiley & Sons, Inc. J Comput Chem 20: 1455–1467, 1999

Keywords: knowledge-based force fields; fold recognition; frozen approximation; protein folding; protein threading; structure prediction

Correspondence to: A. E. Torda; e-mail: Andrew.Torda@anu.edu.au

This article includes Supplementary Material available from the author upon request or via the Internet at ftp.wiley.com/public/journals/jcc/suppmat/20/1455 or http://journals.wiley.com/jcc/

CCC 0192-8651 / 99 / 141455-13

Introduction

Structural biologists routinely dream of being able to predict protein conformation from sequence information alone, but if there is no significant sequence homology to a known structure, there is no reliable recipe. The ambitious may aim for an *ab initio* prediction, trying to search all reasonable conformations in either continuous or discretized space.[1] The more timid may note that a novel sequence is unlikely to have a really new structure, and will probably adopt a fold similar to one already seen.[2] In that case, scanning a library of known structures for the most suitable template will often produce a useful answer. This has led to the popularity of protein-threading approaches for structure prediction,[3,4] and a proliferation of energy or score functions.[5–10]

For protein threading, one wants the best alignment of the sequence on to each member of a library of candidate structures. Each alignment produces a trial structure, and these have to be ranked. The initial alignment is obviously important, but in the most general formulation, allowing for gaps, the sequence–structure alignment problem is NP-complete.[11] This means that every practical approach must use some approximation. The aim of this article is to test a score function simplification that allows a Needleman and Wunsch algorithm[12] to be directly applied for solving the alignment problem. This allows one to generate an optimal alignment in a nonoptimal score function. For ranking the calculated alignments across the library, the simplification can be removed and scores calculated using the best score function available.

There are many approximations possible for sequence to structure alignments. The search space can be reduced by forbidding gaps within secondary structure units. The problem might still be NP-hard, but may be approachable with a method such as Monte Carlo[13,14] or a branch and bound algorithm.[15] To allow a gap of any length, at any position, one may use a dynamic programming algorithm as is commonly employed in sequence comparisons. This requires a matrix with every residue from the sequence at every position in the template and containing its score (or energy) in the field due to its neighbors. Unfortunately, the neighbors have not been aligned, so a direct approach is not possible. One solution is a two-level dynamic programming algorithm[3] where one does not use the real score of a residue interacting with its neighbors, but rather, the best score it could have.[16] A simpler approach is to assume that in similar structures, equivalent positions experience a similar field due to their neighbors. Thus, in the ‘‘frozen approximation,’’ one builds a score matrix by calculating the energy of each sequence residue at each position in the template, but interacting with the residue types of the original template. This means that the structure library is not merely a set of structural scaffolds, but its members actually have a memory of their original composition. One can then use an iterative procedure to attempt to remove the influence of the original template residues. After the initial alignment, template residue identities can be replaced by corresponding residues from the aligned sequence, producing a newly labeled template. On successive iterations, the residues of the template are updated, and the sequence realigned.[17–19]

All of these methods share the philosophy that one score function can be used for both sequence–structure alignment and ranking of alignments from the library. A different approach is to say that one score function may not be best for both purposes. Furthermore, a simplified score function could be built that allowed a sequence–template score matrix to be calculated directly. This function would allow a residue to be scored at any position on a template, without first having to know the exact alignment of its neighbors. That is, all interaction parameters would be of the type AX, BX, CX . . . , where X represents any other type of amino acid. As in a conventional force field, this would require the coordinates of both interaction partners. Unlike a conventional force field, it would require knowing the identity of only one interaction partner. This kind of construction is dubbed a neighbor-nonspecific (NNS) score function. This score function would lose information from the specificity of interactions, but could be optimized so as to minimize the damage. It would only be used for calculating alignments. Once the location of residues has been determined, there is no need for the approximation and the best score function available should be used for ranking the structures.

Aside from issues of interaction partner identity, the score functions used in this work are constructed quite differently to most knowledge-based force fields. There is no assumption that protein structures follow a Boltzmann distribution. Instead, one defines a set of interaction function and parameters that are adjusted so as to optimize score function performance according to some numerical criterion.[20–24] This was used to build a function with a strong ability to distinguish native structures from a very large number (10^7) of misfolded protein-like conformations.[25] The methodology could also be applied to build an alignment (NNS) score function. This may be seen as a kind of averaging where a particle experiences a field due to the average of possible neighbor identities, but is more sophisticated. The parameters are optimized with the averaging present.

This means there are two score functions in this work. The first, using only one particle identity, is referred to as neighbor nonspecific (NNS), and is used for alignment calculation only. The second, relying on the labels of both particles, is termed the neighbor-specific (NS) score function, and has been previously described.[25] Because of the construction methods, the force fields are in arbitrary units and cannot even be expressed in reduced terms of kT. We flaunt this feature, refer to the constructions as score functions, and unashamedly adopt the convention that a higher score is more satisfactory (opposite to energy).

In the following sections, the neighbor-nonspecific (NNS) score function is compared to a more conventional neighbor-specific (NS) score function. This shows the effect of the score function approximations. Next, the NNS is used to calculate sequence to structure alignments that are compared to those from the frozen approximation.[17–19] Finally, problems with convergence in the frozen approximation are demonstrated.

Materials and Methods

PROTEIN SELECTION

Different sets of protein chains were used for deriving score function parameters, building a library of candidate proteins and testing protein fold recognition. Score function parameters were calculated using protein chains from Hobohm and Sander[26] (August 1996 release). A subset was chosen such that each chain had more than 100 residues, all backbone heavy atoms were present, and no two protein chains had more than 25% sequence identity. This resulted in 370 protein chains described previously[25] and in supplementary material (Table S1). A large fold library was built for generating large numbers of alternate alignments. This came from the same source[26] (March 1997 release), and contained all proteins such that no two members shared more than 95% sequence identity, and all heavy backbone atoms were present. This resulted in 1692 proteins listed in supplementary material (Table S2).

Fold recognition was measured using two large datasets from the literature[27,28] described under Results. Each set consisted of pairs of structurally similar proteins with low sequence identity. From each pair, one's sequence is used as a probe, while the structure of the other is hidden in a library of several hundred decoy structures. As well as the original literature, the lists of protein chains are given supplementary material (Table S3 and Table S4). Entries in either set that had been superseded in the July 1997 Protein Data Bank release were replaced by newer versions. Rather than the original set of Fischer et al.[27] with a decoy library of 301 chains, the newer, enlarged set of 320 protein chains was used.[29]

Given the large numbers of proteins, no attempt was made to detect overlap in the sets of proteins used for testing or those used for parameterization and testing. Not only is there the likelihood of some overlap of parameterization and testing sets, but the problem is probably worse because some protein entries are practically identical, but have slightly varying names.

CALCULATION OF SEQUENCE–STRUCTURE ALIGNMENTS

Initial sequence–structure alignments were calculated using an adapted version of the algorithm described by Needleman and Wunsch[12] implemented in the sausage program.[30] Instead of residue/residue scoring matrices, residue/template scores were calculated using the score function described below. Rather than simple gap penalties, geometric penalties for gaps in the sequence were used as described below.

All ranking of models was done after rescoring using only those template locations with an aligned sequence residue.

Sequence iteration followed the method of Godzik et al.[17] Only the highest scoring alignment was saved and returned.

GAP AND INSERTION PENALTIES

A geometric gap penalty, E_{gap} was used for gaps in sequence. From the coordinates of the template, one can calculate the distance between sites in the sequence. The gap penalty, E_{gap}, was calculated according to the distance $d_{C_iN_j}$ between the carbonyl carbon of the first residue, i and the amide nitrogen of the next residue, j

$$E_{\text{gap}} = \begin{cases} 0 & d_{C_iN_j} \le d_0 \\ s(t_i t_j)k_{\text{gap}}\left(d^2_{C_iN_j} - d^2_0\right) & d_0 < d_{C_iN_j} \le d_{\max} \\ s(t_i t_j)k_{\text{gap}}(d^2_{\max} - d^2_0) & d_{\max} < d_{C_iN_j} \end{cases} \tag{1}$$

where d_0 is a reference distance for the carbon–nitrogen distance and $d_{\max}$ is a maximum distance considered, and k_{gap} was a scaling parameter. In all the calculations, d_0 was set at 1.37 Å, slightly longer than the carbon nitrogen bond length. $d_{\max}$ was set at 14 Å. $s(t_i t_j)$ is a switching function to penalize gaps or insertions in regions of secondary structure. t_i and t_j were the secondary structure assignments given by the DSSP program.[31]

$$s(t_i, t_j) = \begin{cases} 1 & t_i \text{ and } t_j \text{ is neither} \\ & \alpha\text{-helix nor } \beta\text{-sheet} \\ k_s & t_i \text{ or } t_j \text{ is } \alpha\text{-helix or } \beta\text{-sheet} \end{cases} \tag{2}$$

The constant k_s is listed with other scaling parameters in Table I.

Insertions in sequence (gaps in template) correspond to more than one sequence residue occupying the same location in the template, and were penalized according to a conventional gap opening/widening scheme. If N_{ins} is the number of inserted residues and $N_{\max}$ is used to limit the maximum penalty size, the insertion penalty E_{ins} was calculated by

$$E_{\text{ins}} = -\begin{cases} 0 \quad N_{\text{ins}} = 0 \\ s(t_i, t_j)k_{\text{ins}}E_{\text{open}} \quad N_{\text{ins}} = 1 \\ s(t_i, t_j)k_{\text{ins}}\left[E_{\text{open}} + E_{\text{wdn}}(N_{\text{ins}} - 1)\right] \\ \quad 1 < N_{\text{ins}} < N_{\max} \\ s(t_i, t_j)k_{\text{ins}}\left[E_{\text{open}} + E_{\text{wdn}}(N_{\max} - 1)\right] \\ \quad N_{\text{ins}} \ge N_{\max} \end{cases} \tag{3}$$

where k_{ins} was a scaling constant. In all calculation, E_{open} the cost of gap opening was set to 0.3, and E_{wdn}, the cost of gap extension was set to 0.01. Scaling constants are given in Table I.

TABLE I.
Parameters Used in Alignment and Rescoring Calculations.

Score Function	k_{gap}	k_{ins}	k_s
Neighbor nonspecific	2000	750	5
Neighbor specific (standard)	250	250	1

SCORE FUNCTION FORM AND PARAMETERIZATION

The functional form and parameterization method for the neighbor-specific force field has been described previously,[25] and is repeated only briefly. Five interaction sites were used for each amino acid, located at the backbone N, C$^\alpha$, C, and O and side-chain C$^\beta$ atoms. A C$^\beta$ interaction site was calculated for glycine residues assuming ideal geometry. There were 20 types of C$^\beta$ particles corresponding to the different residue types, but only one type of each backbone atom. The total score in the neighbor-specific score function for a sequence-structure alignment over N_{res} residues was calculated from

$$E^{\text{spec}}_{\text{tot}} = \sum_{i}^{5N_{\text{res}}} \sum_{j>i}^{5N_{\text{res}}} E^{\text{spec}}_{\text{pair}}(i, j) + \sum_{k}^{N_{\text{res}}} E_{\text{sol}}(k) + E_{\text{gap}} + E_{\text{ins}} \tag{4}$$

where i and j are indices running over all the aligned residues and the summations run over the $5N$ particles. The pair score for particles i and j of residue types t_i and t_j at a topological distance s_{ij} and Cartesian distance of d_{ij} was given by a sigmoidal function

$$E^{\text{spec}}_{\text{pair}}(i, j) = p_{\text{pair}}(s_{ij}, t_i, t_j) \times\left(1 - \tan h\left(w_{\text{pair}}\left(d_{ij} - d^0_{ij}\right)\right)\right) \tag{5}$$

where $p_{\text{pair}}(s_{ij}, t_i, t_j)$ is a parameter determining interaction strength, d^0_{ij} is a reference distance determining the step position and w_{pair} determining the slope of the interaction function. Only three classes of topological distance s_{ij} were considered.

These were $j = i + 2$, $j = i + 3$ and $j \geq i + 4$. The interaction between adjacent residues ($j = i + 1$) was only treated by the gap penalty. The "solvation energy" or particle environment term, E_{sol} was given by a similar function

$$E_{\text{sol}}(i) = p_{\text{sol}}(t_i)\big(1 - \tanh\big(w_{\text{sol}}(n(i) - n^0)\big)\big) \quad (6)$$

where p_{sol} was an adjustable parameter and w_{sol} controlled the slope of this function. $n(i)$ was the number of residues within 5.8 Å (C^α–C^α distance), but separated by more than three residues in the sequence. n^0 was set to 3.

The neighbor nonspecific score function was given by

$$E_{\text{tot}}^{\text{non-spec}} = \sum_{i}^{5N_{\text{res}}} \sum_{j>i}^{5N_{\text{res}}} E_{\text{pair}}^{\text{non-spec}}(i, j) + \sum_{k}^{N_{\text{res}}} E_{\text{sol}}(k) + E_{\text{gap}} + E_{\text{ins}} \quad (7)$$

where $E_{\text{pair}}^{\text{non-spec}}$ was the pairwise score term. This was similar to eq. (5), and depended on the coordinates of both particles i and j, but only the residue type of particle i.

$$E_{\text{pair}}^{\text{non-spec}}(i, j) = p_{\text{pair}}(s_{ij}, t_i) \times \Big(1 - \tanh\Big(w_{\text{pair}}\big(d_{ij} - d_{ij}^0\big)\Big)\Big) \quad (8)$$

where $p_{\text{pair}}(s_{ij}, t_i)$ was a set of parameters for the neighbor-nonspecific score function.

SCORE FUNCTION PARAMETER OPTIMIZATION

Parameters for the neighbor-specific interaction function have been described.[25] A similar approach was used to optimize the parameters for the neighbor-nonspecific score function. First, the z-score for a native sequence–structure pair was defined by

$$z = \frac{\langle \Delta E \rangle}{\sqrt{\langle \Delta E^2 \rangle - (\langle \Delta E \rangle)^2}} \quad (9)$$

where ΔE is the difference of score ($E_{\text{nat}} - E$) between the native sequence–structure and an alternate structure, and the angle brackets denote the arithmetic average over the collection of alternate conformations. For the neighbor-nonspecific score function, these scores were calculated from eq. (7). The alternative conformations were generated from every possible ungapped alignment of the sequence on the coordinates of every parameterization protein of the same or a greater number of residues.

Parameters were then adjusted so as to optimize the score function's ability to distinguish native from nonnative sequence–structure pairs. The adjustable parameters are considered as a vector, $\vec{P}$, on which the score $E_{\text{tot}}^{\text{non-spec}}(\vec{P})$ is linearly dependent. Equation (9) is then expanded in terms of these parameters. For clarity, we do not write the obvious dependence on coordinates. A target function was then defined so as to encapsulate force field quality

$$t(\vec{P}) = \frac{1}{n_{\text{lib}}} \sum_{i}^{n_{\text{lib}}} \Big(z_i(\vec{P}) + 15\Big)^4 \quad (10)$$

where z_i is the z-score of protein sequence i, calculated using the parameterization protein set. The summation runs over all n_{lib} (here 370) proteins in the set. The constants 15 and 4 were determined by trial and error. The target function [eq. (10)] was then maximized using the fast method described previously.[25] This is equivalent to adjusting the force field so as to get the best possible discrimination of native from alternate structures. As in the earlier work, conjugate gradients was used for minimization, but it appears that a global optimum was reached, presumably due to the simple functional forms of eqs. (5), (6), and (8).

For the neighbor-nonspecific score function, $\vec{P}$ consisted of $p_{\text{pair}}(s_{ij}, t_i)$ for each amino acid and atom type at each of the three topological distances, and the 20 $p_{\text{sol}}(t_i)$ parameters. For one topological distance, this meant $4 \times (4 + 1)/2$ backbone–backbone interaction, four backbone–side-chain interactions, and 5×20 backbone–side-chain interactions. Summing over the three topological distances and adding the 20-particle environment terms gave a total of 362 adjustable parameters. For the neighbor-specific force field, 920 parameters were optimized.[25]

STRUCTURE COMPARISON AND MEASURES OF FOLD RECOGNITION

All pairwise structural comparisons were calculated from root-mean-square differences of distance matrices based on backbone C^α coordinates.[32,33] This is often referred to as the distance matrix error or DME,[20,34] and defined by

$$DME = \left(\frac{2}{N(N-1)} \sum_{i<j}^{N} (d_{ij} - d'_{ij})^2 \right)^{1/2} \qquad (11)$$

where the indices i and j run over all the N particles (C^{α} atoms) and d_{ij} and d'_{ij} are the distance between atoms i and j in the first and second structures.

Structural similarity between probe sequences and homologous structures was measured as in Rost et al.[27] Given the length of the probe sequence, L_1, the template structure, L_2, and the number of aligned residues from a structural alignment L_{ali}, one can define a ratio R_{ali}

$$R_{\text{ali}} = \frac{2 \times L_{\text{ali}}}{L_1 + L_2} \qquad (12)$$

Values for R_{ali} were taken from the FSSP database.[36–39] The success of fold recognition was judged by the cumulative percentage of correct folds as used in Rost et al.[27] and earlier defined:[35]

$$Q(R) = 100 \sum_{r=1}^{R} \frac{N_{\text{corr}}(r)}{N_{\text{prot}}} \qquad (13)$$

where $N_{\text{corr}}(r)$ gives the number of correct first-rank folds at rank r and N_{prot} is the number of probe sequences in the test set.

Results

APPROXIMATIONS IN THE NONSPECIFIC SCORE FUNCTION

Neglecting the identity of one member of each interaction pair might be a serious approximation. We can, however, directly measure its effect by comparing scores from the neighbor-specific and neighbor-nonspecific functions. For this comparison, three protein sequences were chosen so as to span range from small (50 residues, 4tgf) to medium (150 residues, 1 sct chain B) and larger (301 residues, 1ezm) structures. Each sequence was then aligned to every member of the fold library using the nonspecific score function and allowing gaps and insertions yielding 3 × 1692 structures. Figure 1 shows the correlation between the NNS and NS scores. The first feature is clear. A high-scoring sequence–structure pair will be favored by either function. A second feature is less obvious. The points do not necessarily fit to a straight line. This is not surprising or disappointing. Each score function is optimized to distinguish native from misfolded structures, but does so using a different balance of terms. Furthermore, the ideal result would be any monotonic relationship, even if very nonlinear.

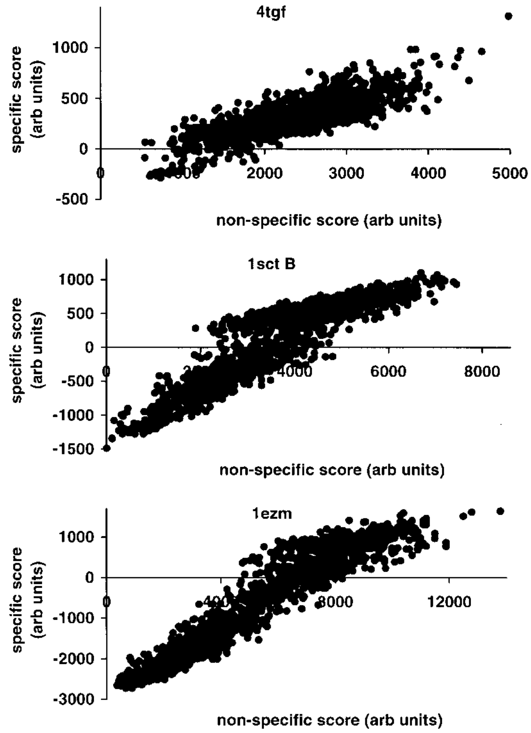


FIGURE 1. Comparison of nonspecific and neighbor-specific score functions. Each of the three sequences was aligned to 1692 templates, and the resulting structures scored using the neighbor-specific and nonspecific functions. Scores are in arbitrary units.

The figures show plots from three protein sequences with gapped alignments, but are typical of the score functions tested on other protein sequences and with ungapped alignments.

ALIGNMENT QUALITY OF NEIGHBOR-NONSPECIFIC SCORE FUNCTION AND FROZEN APPROXIMATION

To compare two sequence–structure alignment methods, one can take some pairs of structurally similar proteins. Within each pair, one is labeled the probe sequence and the other the homologous structure. One can then calculate sequence–structure alignments and compare the quality of the answers produced by the neighbor nonspecific (NNS) score function and the frozen approximation. To minimize debate about benchmarks, protein libraries, bias, and crossvalidation, we have taken a rather large number of pairs from two sources: (a) the benchmark of Fischer et al.[28] and (b) Rost et al.[27] The first is a set of 68 pairs, each with less than 30% sequence identity between the probe and homologue, and where at least half the residues of the larger sequence superimpose on the smaller with a difference of less than 3 Å. The set from Rost et al.[27] consists of 89 probes, but each may have several structural homologues, giving 1003 pairs. Each probe has less than 25% sequence identity to its structural homologues. This set contains some pairs of very similar proteins, but others where there are only small fragments of similarity.

The most direct measure of alignment quality is to see how useful it is for predicting structure. After an alignment, one can build a model for the probe sequence, using the coordinates of the template. This model can be structurally compared to the correct answer (probe native structure). For this comparison, we have calculated the root-mean-square difference of distance matrices, often referred to as the distance matrix error (DME) given by eq. (11). When comparing models from the two alignment methods, we required that they be of similar size, so comparisons were not considered if the two alignment methods produced models differing by more than five residues. This reduced the number of points from 68 to 46 in the Fischer et al.[28] set and from 1003 to 781 in the Rost et al.[27] test set. Small models were also removed from the comparison because these would result in misleading, small DME values. These can arise from large gaps or skewed alignments. Requiring that models have at least 50 residues removed 17 more points from the Rost et al.[27] test set, and had no effect on the other set (all models were more than 60 residues).

The results of this comparison are shown in Figure 2. The most outstanding features is that there is very little difference between the two methods. The solid line on each plot is not a line of best fit, but marks $x = y$. If one method was superior, the points would tend to lie to one side of the line.

One can also view the alignments in a less direct manner. With test data sets, one does know the correct structure for the probe sequence, and this can be superimposed on the template. The predicted alignments can then be compared to this reference alignment. One measures the difference (shift) between each residue's position and where the reference suggests it should be, and then calculates an average shift. The set of Rost et al.[27] has a corresponding set of structural alignments readily available,[36–39] so the average shifts were calculated and are shown in Figure 3. The top plot shows that the alignments vary from good to meaningless for both methods. The bottom plot is an expanded region of the same data set showing shifts of up to 20 residues. Regardless of the alignment method, the results span a range from essentially perfect to quite poor. As in Figure 2, the line $x = y$ is marked on both plots, but one could not claim that the points tend to lie above or below the line.

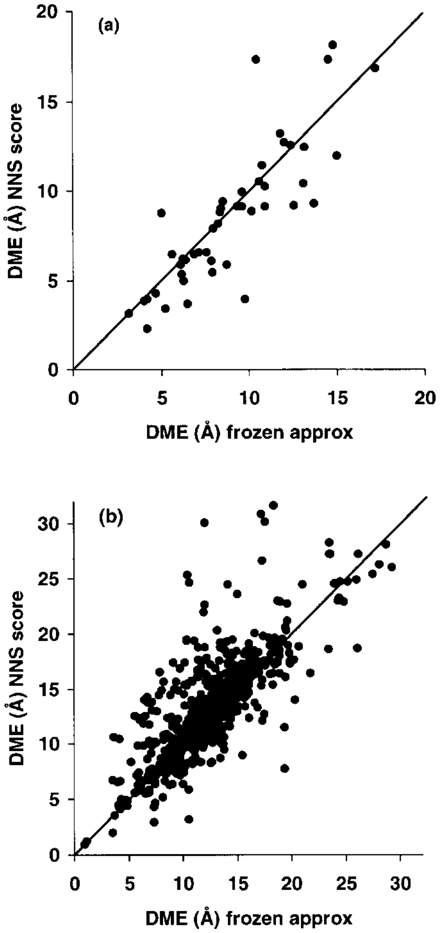


FIGURE 2. Structural comparison of alignments generated using neighbor-nonspecific (NNS) score function and frozen approximation. The distance matrix error (DME) is a measure of the similarity between the model generated by an alignment and the native structure for the probe sequence (a) test set of Fischer et al.[28], (b) set of Rost et al.[27] The line $x = y$ is marked on each plot.

One should not read too much into Figure 3. This treats the alignment as a simple linear prop-

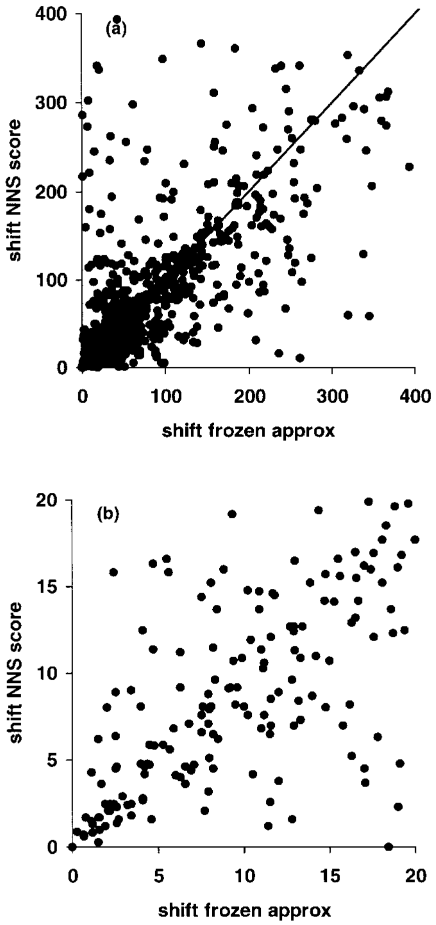


FIGURE 3. Comparison of shifts of alignments generated using neighbor-nonspecific (NNS) score function and frozen approximation. Each axis measures the average shift with respect to a reference structural alignment. (a) shifts of up to 400 residues (b) same data, but with an expanded scale.

erty, whereas one is really interested in the three-dimensional consequences of the alignment. For example, a shift of four residues in a helix may be relatively small compared to a shift of four residues in a β-strand. Second, there will not always be a clear correct structural alignment between nonidentical structures.[40,41] This means that the two alignment methods may produce results that are of equal quality, but one appears better because it is closer to the reference alignment selected by some structure alignment program.

Regardless of these details, there are some clear conclusions. The neighbor-nonspecific score function appears no worse than a more conventional function combined with the frozen approximation. This is remarkable when one considers the functional forms. The neighbor-nonspecific score function models pairwise interactions with just over 20 adjustable parameters for a given topological distance. In contrast, the more conventional score function used in the frozen approximation begins with (20 × 21)/2 corresponding parameters.

SEQUENCE ITERATION

When using the frozen approximation, sequence iteration or "iterative thawing" was performed as described in Godzik et al.[17] This means that an initial alignment of sequence to structure is performed by calculating the score of sequence residues in the field due to the residue types of the template. After this initial alignment, sites on the template can have their residue type replaced by those from the aligned sequence. The sequence can be realigned to this newly labeled template, and this iterative process can be performed a number of times. In the first calculations, it appeared that the runs did not always converge in less than 20 steps. This was investigated in more detail.

In the previous section, there was a large number of probe sequences, each aligned to a relatively small number of homologous structures. In a real structure prediction, one is more likely to take a sequence and attempt alignments to a large number of templates in a library of candidate structures. Most of these will have no significant similarity to the correct answer. This is the approach we took to measure the convergence properties of sequence iteration. A small number of sequences (six) were taken, but aligned to all 1692 members of a structure library. Unlike previous workers, we not only checked for convergence, but iterations were stopped if cycling was detected. That is, some iteration would produce an alignment identical to one already visited. Calculations were limited to 20 iterations, but stopped if there was no improvement of the best score for any 10 successive steps. This last case was classed as divergence (as opposed to convergence), even though the system might cycle or converge given enough iterations.

The results are summarized in Table II. The simplest summary is given by the first two columns, which state the fraction of the calculations in which sequence iteration was useful. A run was classed as successful if, on any iteration, the score rose above the initial value. Thus, a calculation might begin to cycle, yet still produce a better scoring alignment than the starting point. The results vary between proteins, but sequence iteration only appears useful on between a quarter

TABLE II.
Results of Sequence Iteration.

			Overall %		Convergence %		Exceeded Limit %	
Protein[a]	Class[b]	Size	Success	Failure	Converged	Cycled	Diverge	Incomplete
1bjm A	α	162	31	69	10	39	48	3
1cpc A	β	211	36	64	10	57	32	1
1edh	α/β	218	31	69	6	42	49	3
1gta	β	216	24	76	9	38	50	3
1try	β	224	27	73	8	44	46	3
2mnr	α/β	357	8	92	18	38	43	1

[a] PDB acquisition code with chain identifier appended where appropriate.
[b] Fold class taken from Murzin et al.[50]

and third of sequence–structure alignments. The phenomenon of cycling alignments is not insignificant, accounting for anywhere from a third to half the sequence–structure alignments attempted. Perhaps the most striking result is listed under divergence. Between 30 and 50% of sequence iteration calculations actually deteriorate over 10 or more successive sequence iterations.

Another way to judge the utility of the method is to consider all the alignments (6 × 1692) and count the number of times the best result was achieved without iteration, on the first iteration, and so on. This is shown in Figure 4. As expected from Table II, the best number of iterations is usually zero. Some alignments improve with one iteration, fewer with two, and so on. A point not clear from Table II is that even though many iteration converge or cycle, they do not necessarily converge in the sense one usually expects in optimization methods. The alignment may become stable, but it is not necessarily an optimum of any kind.

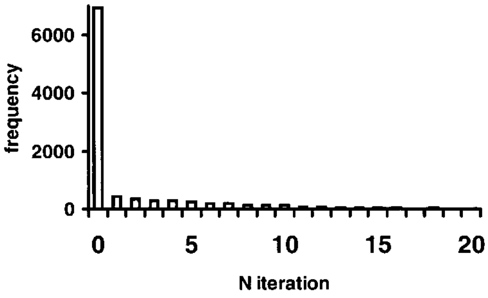


FIGURE 4. Iteration number of best alignment during sequence iteration. Each bar shows the number of times the best alignment score occurred on that iteration. Results are summed over all six sequences, aligned to the protein fold library of 1692 templates.

DIFFERENCE BETWEEN FOLD RECOGNITION AND ALIGNMENT

We have proposed that one may tackle fold recognition using two scoring functions—the first for sequence to structure alignments within a library, and the second for ranking the alignments. One may wonder if the second score function is necessary. Figure 1 suggests that the neighbor-nonspecific (NNS) score function is well correlated with the neighbor-specific score function, and perhaps the simple score function would be sufficient for fold recognition. This can be answered by using both NNS and NS functions for ranking a set of alignments.

In the previous sections, we used known homologues to assess alignment quality. Here, a larger calculation is used based on complete test sets and where the goal is to find structural homologues within a library of decoys. The first set of Fischer et al.[27] had 68 probe sequences with structural homologues hidden in a library of 320 decoys. The second set, from Rost et al.[27] had 89 probes and a library of 723 decoys.

The set of Rost et al.[27] lent itself to further analysis because the sequence/homologue pairs had been labeled according to their degree of structural similarity. A quantity R_{ali}, given by eq. (12), was used. An R_{ali} near 0.9 means that nearly the entire structure are similar, whereas R_{ali} near 0.1 would mean that only small fragments of probe and template were structurally similar.

In total, this meant the fold recognition measurements were performed with both score functions for four sets of proteins: (a) full 89 protein set of Rost et al.,[27] (b) the subset of 70 proteins with $R_{\text{ali}} \geq 0.3$, (c) the subset of 12 proteins with $R_{\text{ali}} \geq 0.8$, and (d) the full set of Fischer et al.[27]

The results of the score function comparison are shown in Figure 5. Following Rost et al.,[27] the cumulative frequency of the first successful prediction $Q(R)$ is plotted, as defined by eq. (13). This measures the rank at which the first correct homologue is detected. For example, a $Q(5)$ of 40 means that, considering all the probes, a correct homologue was found within the first 5 guesses 40% of the time. It only considers the first correct homologue, and does not indicate where the method has been more successful and several homologues have been detected for a probe.

The plots clearly answer the original question. The simplified NNS score function may be used for sequence to structure alignment, but should not be used for overall fold recognition. In every set of proteins, the rankings from the neighbor-specific score function are consistently better than those from the NNS score function.

Somewhat unintentionally, the plots may reveal some of the characteristics of the protein test sets. Qualitatively, it would appear as if the set from Fischer et al.[27] in Figure 5d is closest to the Rost et al.[27] data set with $R_{\mathrm{ali}} \approx 0.5$ in Figure 5b. This may not be coincidence given the selection of the Fischer et al.[27] set required that at least half the residues of the larger protein in each pair readily superimpose on the smaller.

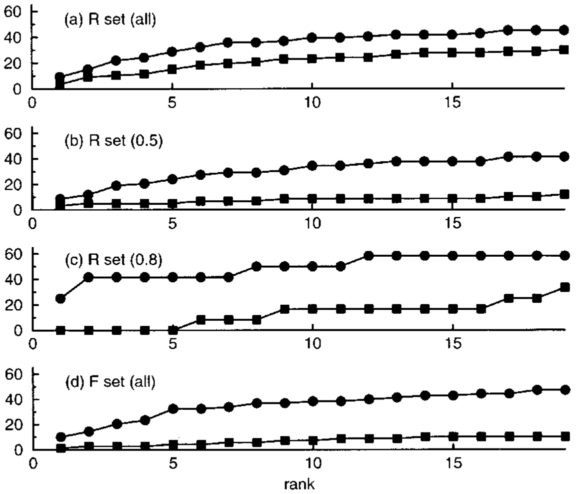


FIGURE 5. Fold recognition, Q (rank) with different force fields and different test sets. In each case, the lower curve (□) shows fold recognition in the neighbor-nonspecific score function. The upper line (●) shows fold recognition in the neighbor specific score function. (a) Data set from Rost et al.,[27] (b) same as above but with $R_{\mathrm{ali}} \geq 0.5$; (c), same as above but with $R_{\mathrm{ali}} \geq 0.8$, (d) data set from Fischer et al.[28]

The plots also reveal some of the fold recognition properties of the score functions, even if the aim has been to look at sequence–structure alignments. The top panel Figure 5a can be directly compared with Rost et al.[27] It would appear that the functions used here are slightly less effective than those of Rost et al.[27] It is not until there is a significant degree of structural overlap with $R_{\mathrm{ali}} \geq 0.7$ or $R_{\mathrm{ali}} \geq 0.8$ that the functions may approach being useful. This is not very different from the findings of Bryant.[42] From Figure 5c, one might say that at this level of structural similarity, there is a one in two chance of finding a correct homologue in the first 5 to 10 guesses.

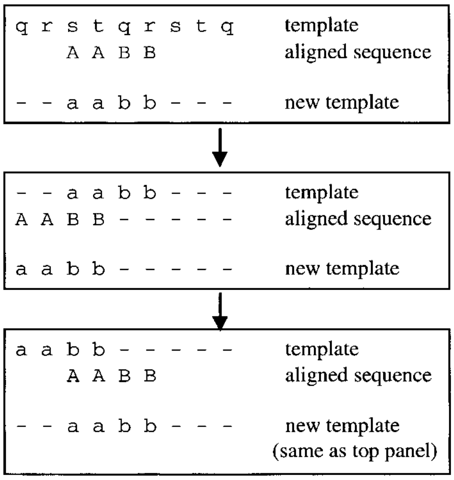
Discussion

From Figures 2 and 3, it would appear that the simplified, neighbor-nonspecific (NNS) score function works as well (or badly) as the frozen approximation for sequence–structure alignment. This result is somewhat disappointing. Until recently, the frozen approximation had its foundations in optimism, intuition, and the hope that environments would be conserved in structurally similar proteins. Zhang et al.[43] have attempted to quantify the validity of some of the assumptions. From their results it may seem remarkable that the method works as well as it does as the first step in many protein-threading packages.

The NNS score function obviously does not do better in its current form, but has some clear advantages. It has an order of magnitude less adjustable parameters than the comparable conventional neighbor-specific score function. From a practical point of view, one can gain a huge time saving compared to sequence iteration. If one needs 2 to 10 iterations to find a final alignment, one can gain 2- to 10-fold speed by using a simplified NNS score function. Finally, alignments with the NNS score function do not use the residue types of the template at all. The method is completely free of what has been referred to as template sequence memory.

Aside from issues of performance, there are some results that should be compared to previous workers. The most surprising of these are the disappointing results produced by sequence iteration and the frozen approximation. One would like to know if the problems, including cycling of alignments, occur with score functions based on Boltzmann statistics or other knowledge-based functions. Godzik et al.[17] found that sequence iteration usually converges in 5 to 10 iterations, whereas less than 20% of the calculations in this work converged in less than 10 iterations. Wilmanns and Eisenberg[19] state that, using their measure of correctness, 75% of initial alignments that begin as more than 50% correct improve with sequence iteration. Because they also had a number of alignments less than 50% correct, this result may be closer to those seen in our work. Unfortunately, it appears that no previous worker has quantified by cycling of alignments. Maybe the problem is not universal. We have certainly found that convergence problems can be alleviated with larger gap penalties (data not shown).

Because qualitative comparison with previous workers is difficult, one might try to reason whether the problems seen here will be more general. In fact, it is easy to create a simple score function and model alignment that demonstrates cycling. Consider a case with residues of types A and B. Like residues repel, but the AB interaction is attractive, rather like residues of opposite charge. A sequence AABB is aligned to a template whose original residues are qrst. This is shown in Figure 6. To complete the example, say that A residues have a preference for a secondary structure such as α-helix, and that this is the structure of the first few residues of the template. Using the frozen approximation, some alignment is produced, as shown in the top panel. For sequence iteration, the residues of the template are then replaced by the aligned sequence, generating the newly labeled template shown on the third line of the top panel. At the next iteration, the sequence is aligned so as to maximize the AB interaction and also the secondary structure preference of the A residues. In the third panel, the sequence is again aligned, and the template residues replaced. The new template, however, is identical to the template of the top panel. This model system will cycle forever without convergence.



```
q r s t q r s t q     template
    A A B B           aligned sequence

- - a a b b - - -     new template
```

```
- - a a b b - - -     template
A A B B - - - - -     aligned sequence

a a b b - - - - -     new template
```

```
a a b b - - - - -     template
    A A B B           aligned sequence

- - a a b b - - -     new template
                      (same as top panel)
```

FIGURE 6. Sequence iteration and cycling of alignments. In each panel, the top line (lower case) shows the current identity of template residues, and the line below it (upper case) the alignment of the sequence (upper case). The third line shows the new identities assigned to the template residues after that iteration.

This very simple model system cycles with a period of two iterations, and moving to a more realistic score function allows one to generate more complicated cycling patterns. In a larger protein, different parts of the molecule cycle simultaneously with different periods. More generally, convergence of sequence iteration is based on assumption about the score/energy surface on which the residues move. That is, the residues of an incorrect alignment should have the effect of moving their neighbors towards more correct locations. This may or may not be the case and will certainly vary between score function as they determine the shape of the "energy" surface. It may well be the case that the score functions used in this work are particularly badly suited to sequence iteration. They have been optimized to distinguish quite sharply between native and incorrect structures, rather than gradually rank the various nonnative structures [eq. (10)].

Aside from issues of convergence in sequence iteration, parameters have more general effects. Gap penalties can be tuned to produce better results for the more distant homologues and produce an apparently better version of Figure 6a (data not shown). This, however, would be at the expense of performance on other protein sets. In general, one may want to use smaller gap penalties when protein structures are less close and a reasonable alignment requires more or larger gaps. This is directly analogous to the use of different gap penalties for sequence alignment, depending on the expected degree of sequence similarity.[44]

From this work, it may seem as if it were a totally new idea to separate sequence to structure alignment from overall fold recognition. This is obviously not the case. It has been proposed that one may need one set of gap penalties for alignments and another for ranking the resulting models,[45] and others have shown that different score function will be best for different problem domains.[46] With the machinery here, one can demonstrate that alignment and raking calculation really are different problem domains. One can compare the distribution of scores/energies encountered at each stage. Considering alignments, it appears that the energies of alternative (decoy) alignments do have a Gaussian-like distribution.[25] Presumably, they do in other workers' score functions because this is a requirement for the validity of the often quoted *z*-scores.[13,18,47,48] After calculating the best alignment for a sequence to each member of a template library, one has a different distribution. The sequence of 1try was aligned to each of 1692 structures in the large template library, and the distribution of scores is shown in Figure 7. This is nothing like a Gaussian distribution. Clearly, during alignment and final ranking, a score function is working in different domains. The choice of 1try was quite arbitrary, and the skewed distributions are similar for all sequences examined.

The most important question raised and not answered here is the degree to which the findings are applicable to other score functions and force fields used for fold recognition. In principle, there is no reason why a Boltzmann-based score function could not be built using only one interaction partner's identity. It remains to be seen what quality of alignments would be produced.

Finally, it may be that the discrimination criterion used in score function construction is not ideal. On simple model systems, the correct alignment may not correspond to the optimum from the force field.[49] This means that a future task is to pursue separate alignment and ranking score functions, but with more appropriate construction methods.

The "sausage" program and parameter sets used in this work are available from ftp://ftp.rsc.anu.edu.au/pub/torda/sausage/README.

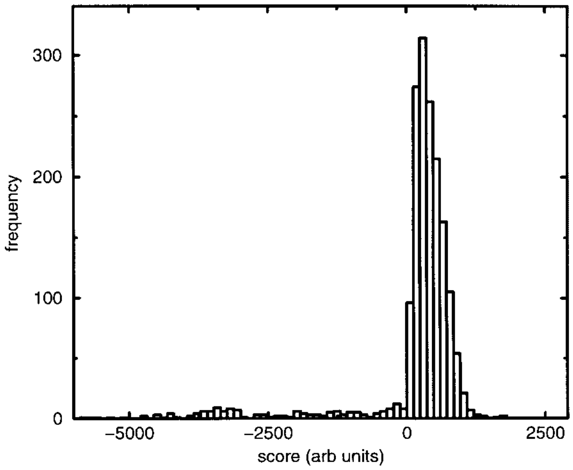


FIGURE 7. Distribution of scores of aligned models for 1try sequence. Alignments were calculated in the neighbor-nonspecific force field and the resulting alignments scored in the neighbor-specific score function. Score is given in arbitrary units with less favorable scores being more negative, and frequency is the number of times a score was observed within the 1692 models.

Acknowledgments

We thank Dan Ayers for a heroic perl script and Adrian Cootes for acronymic advice.

References

1. Scheraga, H. A. Biophys Chem 1996, 59, 329.
2. Holm, L.; Sander, C. Proteins 1994, 19, 165.
3. Jones, D T.; Taylor, W. R.; Thornton, J. M. Nature 1992, 358, 86.
4. Sippl, M. J.; Weitckus, S. Proteins 1992, 13, 258.
5. Böhm, G. Biophys Chem 1996, 59, 1.
6. Jernigan, R. L.; Bahar, I. Curr Opin Struct Biol 1996, 6, 195.
7. Jones, D. T.; Thornton, J. M. Curr Opin Struct Biol 1996, 6, 210.
8. Sippl, M. J. Curr Opin Struct Biol 1995, 5, 229.
9. Sippl, M. J.; Flöckner, H. Structure 1996, 4, 15.
10. Torda, A. E. Curr Opin Struct Biol 1997, 7, 200.
11. Lathrop, R. H. Protein Eng 1994, 7, 1059.
12. Needleman, S. B.; Wunsch, C. D. J Mol Biol 1970, 48, 443.
13. Bryant, S. H.; Lawrence, C. E. Proteins 1993, 16, 92.
14. Madej, T.; Gibrat, J.-F.; Bryant, S. H. Proteins 1995, 23, 356.
15. Lathrop, R. H.; Smith, T. F. J Mol Biol 1996, 255, 641.
16. Taylor, W. R. J Mol Biol 1997, 269, 902.
17. Godzik, A.; Kolinski, A.; Skolnick, J. J Mol Biol 1992, 227, 227.
18. Sippl, M. J. J Comput Aided Mol Des 1993, 7, 473.
19. Wilmanns, M.; Eisenberg, D. Protein Eng 1995, 8, 627.
20. Crippen, G. M. J Mol Biol 1996, 260, 467.
21. Koretke, K. K.; Luthey−Schulten, Z.; Wolynes, P. G. Protein Sci 1996, 5, 1043.
22. Mirny, L. A.; Shakhnovich, E. I. J Mol Biol 1996, 264, 1164.
23. Ulrich, P.; Scott, W.; van Gunsteren, W. F.; Torda, A. E. Proteins 1997, 27, 367.
24. Hao, M.-H. Scheraga, H. A. Proc Natl Acad Sci USA 1996, 93, 4984.
25. Huber, T.; Torda, A. E. Protein Sci 1998, 7, 142.
26. Hobohm, U.; Sander, C. Protein Sci 1994, 3, 522.
27. Rost, B.; Schneider, R.; Sander, C. J Mol Biol 1997, 270, 471.
28. Fischer, D.; Elofsson, A.; Rice, D.; Eisenberg, D. In Biocomputing: Proceedings of the 1996 Pacific Symposium; Hunter, L.; Klein, T., Eds., World Scientific Publishing Co.: Singapore, 1996, p. 300.

29. Fischer, D.; http://www.doe-mbi.ucla.edu/people/fischer/BENCH/benchmark1.html (1998).
30. Huber, T.; Rusell, A. J.; Ayers, D. J.; Torda, A. E. Bioinformatics 1999, in press and http://www.rsc.anu.edu.au/~torda/sausage.html and ftp://ftp.rsc.anu.edu.au/pub/torda/sausage/README
31. Kabsch, W.; Sander, C. Biopolymers 1983, 22, 2577.
32. Rooman, M. J; Rodriguez, J.; Wodak, S. J. J Mol Biol 1990, 213, 327.
33. Levitt, M. J Mol Biol 1983, 168, 621.
34. Havel, T. F. Biopolymers 1990, 29, 1565.
35. Rost, B. http://www.embl-heidelberg.de/~rost/Papers/PreTopits96.html (1996).
36. Holm, L.; Sander, C. Nucleic Acids Res 1996, 24, 206.
37. Holm, L.; Sander, C. Nucleic Acids Res 1994, 22, 3600.
38. Holm, L.; Sander, C. Nucleic Acids Res 1997, 25, 231.
39. Holm, L.; Sander, C. Nucleic Acids Res 1998, 26, 316.
40. Godzik, A. Protein Sci 1996, 5, 1325.
41. Feng, Z. K.; Sippl, M. J. Fold Des 1996, 1, 123.
42. Bryant, S. H. Proteins 1996, 26, 172.
43. Zhang, B.; Jaroszewski, L.; Rychlewski, L.; Godzik, A. Fold Des 1997, 2, 307.
44. Henikoff, S. Curr Opin Struct Biol 1996, 6, 353.
45. Westhead, D. R.; Collura, V. P.; Eldridge, M. D.; Firth, M. A. Li, J.; Murray, C. W. Protein Eng 1995, 8, 1197.
46. Park, B. H.; Huang, E. S.; Levitt, M. J Mol Biol 1997, 266, 831.
47. Kocher, J.-P. A.; Rooman, M. J.; Wodak, S. J. J Mol. Biol 1994, 235, 1598.
48. Jones, D. T.; Miller, R. T.; Thornton, J. M. Proteins 1995, 23, 387.
49. Crippen, G. M. Folding Des 1997, 2, S 58.
50. Murzin, A. G.; Brenner, S. E.; Hubbard, T.; Chothia, C. J Mol Biol 1995, 247, 536.