S.L. Salzberg, D.B. Searls, S. Kasif (Eds.), *Computational Methods in Molecular Biology*
© 1998 Elsevier Science B.V. All rights reserved

CHAPTER 13

# THREADER: protein sequence threading by double dynamic programming

David Jones

*Department of Biological Sciences, University of Warwick (UK)*

## 1. Introduction

The prediction of protein tertiary structure from sequence may be expressed symbolically by expressing the folding process as a mathematical function

$$\boldsymbol{C} = F(\boldsymbol{S}),$$

where

$$\boldsymbol{S} = [s_1, s_2, \ldots, s_n], \qquad \boldsymbol{C} = [\theta_1, \theta_2, \ldots, \theta_{3n-2}], \qquad s \in \{\text{Ala}, \text{Arg}, \ldots, \text{Val}\}.$$

In this case the main chain conformation of the protein chain $\boldsymbol{S}$ is represented as a vector of main chain torsion angles $\boldsymbol{C}$, with the chain itself being defined as a vector of elements corresponding to members of the set of 20 standard amino acids. The folding process is therefore defined as a function which takes an amino acid sequence and computes from it a sequence of main chain torsion angles. The choice of representation of the folded chain conformation in torsion space is arbitrary, and the problem can just as readily be expressed in terms of relative orthogonal 3D coordinates, or with some indeterminacy in chirality, interatomic distances.

The protein folding problem can thus be considered a search for the folding function $F$. It is probable, however, that no simple representation of the folding function exists, and that even if the function exists in any form whatsoever, the only device capable of performing the required function evaluation is the protein chain itself. Conceptually, the simplest way to arrange for a protein sequence to code for its own native 3D structure is to arrange for the native structure to be the global minimum of the protein chain's free energy. The folding process is therefore transformed into an energy function minimization process, where the energy function could take as input the protein sequence vector $\boldsymbol{S}$, and the vector of torsion angles $\boldsymbol{C}$. Given a particular sequence $\boldsymbol{S}$, the folding process is therefore transformed into a *search* through the set of all corresponding vectors of torsion angles $\boldsymbol{C}$ for the minimum of an energy function $E$, where $E$ is defined thus:

$$E(\boldsymbol{S}, \boldsymbol{C}_{\text{native}}) < E(\boldsymbol{S}, \boldsymbol{C}_{\text{non-native}}).$$

The exact form of this energy function is as yet unknown, but it is reasonable to assume that it would incorporate terms pertaining to the types of interactions observed in protein structures, such as hydrogen bonding and van der Waals effects. The conceptual simplicity of this model for protein folding stimulated much research into *ab initio* tertiary structure prediction. A successful *ab initio* approach necessitates the solution of two problems. The first problem to solve is to find a potential function for which the above inequality at least generally holds. The second problem is to construct an algorithm capable of finding the global minimum of this function. To date, these problems remain essentially unsolved, though some progress has been made, particularly with the construction of efficient minimization algorithms.

It is unlikely that proteins really locate the global minimum of a free energy function in order to fold into their native conformation. The case against proteins searching conformational space for the global minimum of free energy was argued by Levinthal [1]. The *Levinthal paradox,* as it is now known, can be demonstrated fairly easily. If we consider a protein chain of $N$ residues, we can estimate the size of its conformational space as roughly $10^N$ states. This assumes that the main chain conformation of a protein may be adequately represented by a suitable choice from just 10 main chain torsion angle triplets for each residue. In fact, Rooman et al. [2] have shown that just 7 states are sufficient. This of course neglects the additional conformational space provided by the side chain torsion angles, but is a reasonable rough estimate, albeit an underestimate. The paradox comes from estimating the time required for a protein chain to search its conformational space for the global energy minimum. Taking a typical protein chain of length 100 residues, it is clear that no physically achievable search rate would enable this chain to complete its folding process. Even if the atoms in the chain were able to move at the speed of light, it would take the chain around $10^{82}$ seconds to search the entire conformational space, which compares rather unfavorably to the estimated age of the Universe ($10^{17}$ seconds).

Clearly proteins do not fold by searching their entire conformational space. There are many ways of explaining away Levinthal's paradox. A highly plausible mechanism for protein folding is that of encoding a *folding pathway* in the protein sequence. Despite the fact that chains of significant length cannot find their global energy minimum, short chain segments (5–7 residues) could quite easily locate their global energy minimum within the average lifetime of a protein, and it is therefore plausible that the location of the native fold is driven by the folding of such short fragments [3]. Levinthal's paradox is only a paradox if the free energy function forms a highly convoluted energy surface, with no obvious downhill paths leading to the global minimum. The folding of a short fragment can be envisaged as the traversal of a small downhill segment of the free energy surface, and if these paths eventually converge on the global energy minimum, then the protein is provided with a simple means of rapidly locating its native fold.

One subtle point to make about the relationship between the minimization of a protein's free energy and protein folding is that the native conformation need not correspond to the global minimum of free energy. One possibility is that the folding pathways initially locate a local minimum, but a local minimum which provides stability for the average lifetime of the protein. In this case, the protein in question would always be observed with a free energy slightly higher than the global minimum *in vivo*, but would eventually locate its global minimum if isolated and left long enough *in vitro* – though the location of the global minimum could take many years. Thus, a biologically active protein could in fact be in a *metastable* state, rather than a stable one.

## 2. A limited number of folds

Many fragments of evidence point towards there being a limited number of *naturally occurring* protein folds. If we consider a chain of length 50 residues we might naively calculate the number of possible main chain conformations as $7^{50}$ ($\approx 10^{42}$). Clearly most of these conformations will not be stable folds, and many will not be even physically possible. In order to form a compact globular structure a protein chain necessarily has to form regular secondary structures [4,5], and it is this constraint, along with the constraints imposed from a requirement to effectively pack the secondary structures formed that limit the number of stable conformational states for a protein chain. In addition to the constraints imposed from physical effects on protein stability, there are also evolutionary constraints on the number of occurring folds. Where do new proteins come from? The answer according to Doolittle [6] is of course from other proteins. In other words the folding patterns we observe today are the result of the evolution of a set of ancestral protein folds.

If the number of possible folds is limited, then this fact should be apparent in the presently known protein structures. Do folds recur in apparently unrelated proteins? The answer appears to be a definite "yes" [4]. Reports of these "fold analogies" are becoming more and more common in the literature, though whether this is due to a real saturation effect where the probability of the fold of a newly solved structure matching an existing one increases due to the increase in the number of known folds, or whether this is simply due to an increased awareness of the possibility (and the increased use of structural comparison programs) is a matter of debate.

A limited number of folds and the recurrence of folds in protein which share no significant sequence similarity offer a "short-cut" to protein tertiary structure prediction. As already described, it is impractical to attempt tertiary structure prediction by searching a protein's entire conformational space for the minimum energy structure, but if we know that there could be as few as 1000 possible protein folds [7,8], then the intelligent way to search a protein's conformational space would be to simply consider only those regions which correspond to this predefined set. This is analogous to the difference between an exam requiring the writing of an essay and an exam requiring multiple-choice questions to be answered. Clearly a person with no knowledge of the subject at hand has a much greater chance of achieving success with the multiple-choice paper than with the essay paper.

Suppose we had derived a practical potential function for which the native conformational energy was lower than that of any other conformation, and that we had identified $M$ possible chain folds, then we would have the basis of a useful tertiary structure prediction scheme. In order to predict the conformation of a given protein chain $\boldsymbol{S}$, the chain would be folded into each of the $M$ known chain conformations ($C_1, ..., C_M$), and the energy of each conformation calculated. The predicted chain conformation would be the conformation with the lowest value of the potential function. The term generally applied to schemes of this type is *fold recognition*, where instead of trying to predict the fold of a protein chain *ab initio*, we attempt to recognize the correct chain fold from a list of alternatives.

Figure 1 shows an outline of the fold recognition approach to protein structure prediction, and identifies three clear aspects of the problem that need consideration: a

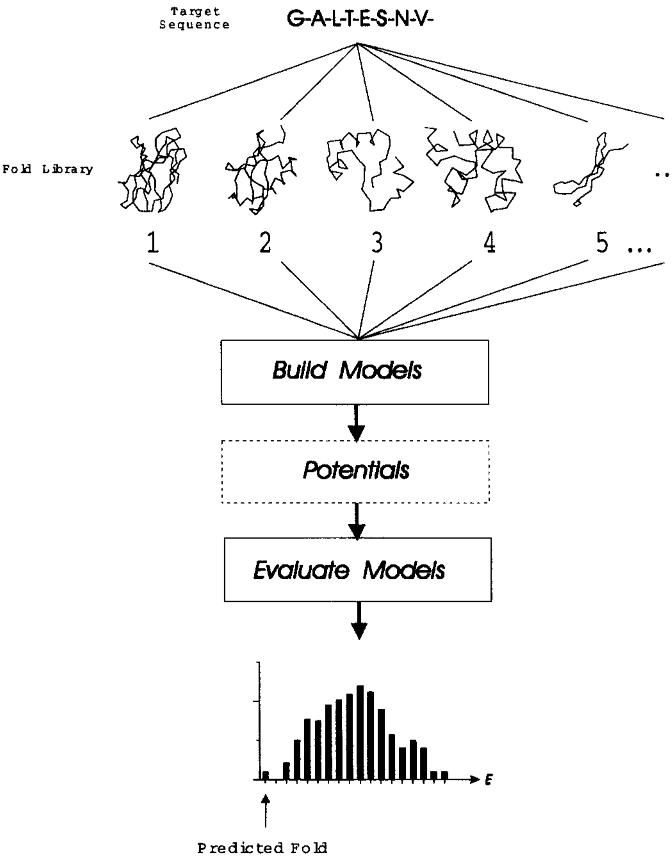

Fig. 1.

fold library, a method for modelling the object sequence on each fold, and a means for assessing the goodness-of-fit between the sequence and the structure.

## *3. The fold library*

A suitable representative library of folds needs to be found. These folds can be observed in crystal structures, NMR structures, or even theoretical model structures. If the library is limited to observed structures then the method will evidently be capable of recognizing only previously observed folds. As has been already discussed, the frequency of occurrence of similar folds between proteins sharing no significant sequence similarity would seem to indicate that creating a library entirely out of known folds is perfectly reasonable. A possible future development for fold recognition might, however, involve the generation of putative model folds, based on folding rules derived from the known structures. Several groups have already attempted the generation of putative folds with some success [9,10], however, the structures created were only very approximate models of real proteins.

## 4. *The modelling process*

The central process in structure-based fold recognition is the fitting of a given sequence onto a structural template. One way of visualizing this process is to imagine the side chains of the object protein being fitted onto the backbone structure of the template protein. This process is of course almost identical to the process of comparative modelling. The standard modelling process consists of three basic steps. Firstly, at least one suitable homologous template structure needs to be found. Secondly, an optimal alignment needs to be generated between the sequence of the template structure (the source sequence) and the sequence of unknown structure (the object sequence). Thirdly, the framework structure of the template is "mapped" onto the object sequence. After several stages of energy minimization, the model is ready for critical evaluation.

Each step in the modelling process has its associated problems, though the first two steps are the most critical overall. Evidently, if no homologous structure can be found, the process cannot even be started, and even when a homologous structure is available (perhaps selected on the basis of functional similarity), the degree of homology may be so low as to render the alignment of the sequences impossible by normal means ("by eye" or by automatic alignment). More recently, pattern matching methods have been developed which offer far greater sensitivity than that offered by simple pairwise sequence alignment [12–15]. These methods in one way or another generate a consensus pattern based on the multiple alignment of several homologous sequences. For example, a globin template [14] may be constructed by aligning the many available globin sequences against a known globin structure, identifying the conserved amino acid properties at each position in the template. Though these methods are capable of inferring reasonably distant homologies, allowing, for example, the modelling of HIV protease based on the aspartyl proteinases [16], they are limited by their dependence on the availability of several homologous sequences, and on the ability of multiple alignment algorithms to successfully align them.

The previously described methods for detecting homology work by increasing the sensitivity of standard sequence comparison algorithms. The general assumption is that some residual sequence similarity exists between the template sequence and the sequence under investigation, which is often not the case. Clearly, therefore, the ideal modelling method would not make this assumption, and work with cases where there is no detectable sequence similarity between the object sequence and the source protein.

A method capable of aligning a sequence with a structural template without reference to the sequence of the template protein is clearly the goal here, but for reasons that will be discussed later, this is a computationally hard problem.

## 5. *Evaluating the models*

The lack of ability of standard atomic force-fields in the detection of misfolded proteins was first demonstrated by Novotny et al. [18]. Their test problem was simple, and yet serves as a good illustration. In this study, the sequences of myohemerythrin and an immunoglobulin domain of identical length were swapped. Both the two native structures, and the two “misfolded” proteins were then energy minimized using the CHARMm [19] force-field. The results were somewhat surprising in that it was impossible to distinguish between the native and misfolded structures on the basis of the calculated energy sums. Novotny et al. realized that the reason for this failure was the neglect of solvation effects in the force-field. In a later study [20], the force-field was modified to approximate some of the effects of solvent and in this case the misfolded structures could be identified reasonably well. The work of Novotny et al. encouraged several studies into effective methods for evaluating the correctness of protein models, which will now be briefly reviewed.

Eisenberg and McLachlan [21] distinguished correct models from misfolded models by using an elegantly simple solvation energy model alone. By calculating a solvation free energy for each amino acid type and calculating the degree of solvent accessibility for each residue in a given model structure, the correctly folded models were clearly distinguished from the misfolded.

Baumann et al. [22] also used a solvation term to recognize misfolded protein chains, along with a large number of other general statistical properties of sequences forming stable protein folds. Holm and Sander [23] have proposed another solvation model, which appears to be very able at detecting misfolded proteins, even those proteins which have shifts of their sequence on their correct native structure. Interestingly enough a sequence–structure mismatch can quite easily occur not just in theoretically derived models, but even in crystallographically derived models. For example one of the xylose-isomerase structures in the current Brookhaven database has in part a clearly mistraced chain. Such errors can be detected by use of a suitable solvation-based model evaluation procedure.

A very widely known method for testing the overall quality of a protein model is that proposed by Lüthy et al. [24], who used a rather more precise definition of residue environment to assess models. This method will be discussed more fully later.

## 6. *Statistically derived pairwise potentials*

Several groups have used statistically derived pairwise potentials to identify incorrectly folded proteins. Using a simplified side chain definition, Gregoret and Cohen [5] derived a contact preference matrix and attempted to identify correct myoglobin models from a set of automatically generated models with incorrect topology, yet quite reasonable core packing.

Hendlich et al. [25] used potentials of mean force, first described by Sippl [26], not only to correctly identify the misfolded protein models of Novotny and Karplus [18], but also to identify the native fold of a protein amongst a large number of decoy conformations generated from a database of structures. In this latter case, the sequence of interest was fitted to all contiguous structural fragments taken from a library of highly resolved structures, and the pairwise energy terms summed in each case. For example, consider a protein sequence of 100 residues being fitted to a structure of length 200 residues. The structure would offer 101 possible conformations for this sequence, starting with the sequence being fitted to the first 100 residues of the structure, and finishing with the sequence being fitted to the last 100. Taking care to eliminate the test protein from the calculation of potentials, Hendlich et al. [25] correctly identified 41 out of 65 chain folds. Using a factor analysis method, Casari and Sippl [27] found that the principal component of their potentials of mean force behaved like a hydrophobic potential of simple form. This principal component potential alone is found to be almost as successful as the full set of potentials in identifying correct folds.

In a similar study to that performed by Hendlich et al. [25], Crippen [28] used simple discrete contact potentials to identify a protein's native fold from all contiguous structural fragments of equal length extracted from a library of highly resolved structures. The success rate (45 out of 56) was marginally higher than that of Hendlich et al. [25] due to the fact that the contact parameters in this case were optimized against a "training set" of correct and incorrect model structures. Maiorov and Crippen [29] improved upon these results using a continuous contact potential, with the new contact function correctly identifying virtually all chain folds defined as being "compact".

Both the work of Hendlich et al. and Crippen demonstrates a very restricted example of fold recognition, whereby sequences are matched against suitably sized contiguous fragments in a template structure. A much harder recognition problem arises when more complex ways of fitting a sequence to a structure are considered i.e. by allowing for relative insertions and deletions between the object sequence and the template structure. Suitable treatment of insertions and deletions is essential to a generalized method for protein fold recognition.

### *6.1. Ponder and Richards (1987)*

The first true example of a fold recognition attempt was the template approach of Ponder and Richards [30] where they concerned themselves with the inverse folding problem. Ponder and Richards tried to enumerate sequences that could be compatible with a given backbone structure. The evaluation potential in this case was a simple van der Waals potential, and so models were effectively scored on the degree of overlap between side chain atoms. A further requirement was for the core to be well-packed, which was achieved by considering the conservation of side chain volume. In order to fit the side chains of a given sequence onto the backbone an exhaustive search was made through a "rotamer library" of side chain conformations. If after searching rotamer space the side chains could not be fitted successfully into the protein core, then the sequence was deemed incompatible with the given fold. As a sensitive fold recognition method, however, this method was not successful. Without allowing for backbone shifts, the packing requirement of a given protein backbone was found to be far too specific. Only sequences very similar to the native sequence could be fitted successfully to the fixed backbone.

### *6.2. Bowie et al. (1990)*

A more successful attempt at fold recognition was made by Bowie et al. [31]. The first stage of this method involves the prediction of residue accessibility from multiple sequence alignments. In essence, alignment positions with high average hydrophobicity and high conservation are predicted to be buried and relatively polar variable positions predicted to be exposed to solvent. The degree of predicted exposure at each position of the aligned sequence family is then encoded as a string. This string is then matched against a library of similarly encoded strings, based, however, not on predicted accessibilities but on *real* accessibilities calculated from structural data. Several successful recognition examples were demonstrated using this method. Of particular note was the matching of an aligned set of Ef Tu sequences with the structure of flavodoxin. The similarity between Ef Tu and flavodoxin is not readily apparent even from structure [32] and so this result is really quite impressive.

### *6.3. Bowie et al. (1991)*

Bowie, Lüthy and Eisenberg [33] attempted to match sequences to folds by describing the fold not just in terms of solvent accessibility, but in terms of the *environment* of each residue location in the structure. In this case, the environment is described in terms of local secondary structure (3 states: $\alpha$, $\beta$ and coil), solvent accessibility (3 states: buried, partially buried and exposed), and the degree of burial by polar rather than apolar atoms. The environment of a particular residue defined in this way tends to be more highly conserved than the identity of the residue itself, and so the method is able to detect more distant sequence–structure relationships than purely sequence based methods. The authors describe this method as a 1D–3D profile method, in that a 3D structure is encoded as a 1D string of amino acids, which can then be aligned using traditional dynamic programming algorithms (e.g. ref. [11]). Bowie et al. have applied the 1D–3D profile method to the inverse folding problem and have shown that the method can indeed detect fairly remote matches, but in the cases shown the hits have still retained some sequence similarity with the search protein, even though in the case of actin and the 70 kD heat-shock protein the sequence similarity is very weak [34]. Environment-based methods appear to be incapable of detecting structural similarities between the most divergent proteins, and between proteins sharing a common fold through probable convergent evolution – environment only appears to be conserved up to a point. Consider a buried polar residue in one structure that is found to be located in a polar environment. Buried polar residues tend to be functionally important residues, and so it is not surprising then that a protein with a similar structure but with an entirely different function would choose to place a hydrophobic residue at this position in an apolar environment. A further problem with environment-based methods is that they are sensitive to the multimeric state of a protein. Residues buried in a subunit interface of a multimeric protein will not be buried at an equivalent position in a monomeric protein of similar fold. In fact, the above authors went on to use their method to successfully evaluate protein models [24], and demonstrated that the method was capable of detecting a previously identified chain tracing error in a structure solved in their own laboratory.

### *6.4. Finkelstein and Reva (1991)*

Finkelstein and Reva [35] used a simplified lattice representation of protein structure for their work on fold recognition, where the problem they considered was that of matching a sequence to one of the 60 possible 8-stranded $\beta$-sandwich topologies. Each strand has 3 associated variables: length, position in the sequence and spatial position in the lattice $Z$ direction. The force-field used by Finkelstein and Reva includes both short-range and long-range components, both based on physical terms rather than statistically derived terms. The short-range component is simply based on the beta-coil transition constants for single amino acids, similar in many respects to the standard Chou–Fasman propensities [36]. The long-range interaction component has a very simple functional form. For a pair of contacting residues, it is defined simply as the sum of their solvent transfer energies as calculated by Fauchere and Pliska [37].

The configurational energy of the 8 strands in this simple force-field is minimized by a simple iterative algorithm. At the heart of the method is a probability matrix (a 3-dimensional matrix in this case) for each of the strands, where each matrix cell represents one triplet of the strand variables i.e. length, sequence position and spatial position. The values in each cell represent the probability of observing the strand with the values associated with the cell. The novel aspect of this optimization strategy is that the strands themselves do not physically move in the force-field, only the probabilities change. At the start of the first iteration the strand positional probabilities are assigned some arbitrary value, either all equal, or set close to their expected values (the first strand is unlikely to be positioned near the end of the sequence for example). A new set of probabilities is then calculated using the current mean field and the inverse Boltzmann equation (see later for more about the inverse Boltzmann equation). As more iterations are executed it is to be hoped that most of the probabilities will collapse to zero, and that eventually a stable "self-consistent" state will be reached. Finkelstein and Reva found that the most probable configurations corresponded to the correct alignment of the 8-stranded model with the given sequence, and that when the process was repeated for each of the 60 topologies, in some cases the most probable configuration of the native topology had the highest probability of all.

The simplicity of the lattice representation used here and the uncomplicated force-field are probably critical to the success of this method. A more detailed interresidue potential would prevent the system from reaching a self-consistent state, and would be left either in a single local minimum or more likely oscillating between a number of local minima. In addition, whilst it is quite practical to represent β-sheets on a lattice, it is not clear how α-helices could be reasonably represented, though in later work, the authors have used highly simplified real 3D protein structures as pseudo-lattices.

## 7. *Optimal sequence threading*

The method described in this chapter has something in common both with the method of Bowie, Lüthy and Eisenberg, and that of Finkelstein and Reva. Despite the obvious computational advantages of using residue environments, it is clear that the fold of a protein chain is governed by fairly specific protein–protein and protein–solvent atomic interactions. A given protein fold is therefore better modelled in terms of a "network" of pairwise interatomic energy terms, with the structural role of any given residue described in terms of its interactions. Classifying such a set of interactions into one environmental class such as "buried alpha-helical" will inevitably result in the loss of useful information, reducing the *specificity* of sequence–structure matches evaluated in this way. The main difficulty in the use of environments alone for recognizing protein folds is that helices look like other helices, and strands like other strands. A sequence that folds into one helix of particular structure, will probably easily fold into any other helix of similar

```
    APRKF----------------FVGGNWKMNGKRKSLGELIHTLDGAKLSADTEVVCGAPS
TIM *9992----------------0000103032*8*400*10*61262*957*261000002
    .....----------------PPPPP..B...HHHHHHHHHHHHH....SS.PPPPP..T

    ..HHHHH....S.......SSPPPPP..----SHHHHHHHHHHHTTT..S--PPPPP.S.
LDH *7*****96*********3*61000000----443020006200*77104--10000299
    ATLKDKLIGHLATSQEPRSYNKITVVGV----GAVGMACAISILMKDLAD--EVALVDVM

    IYLDFARQKLDAK---------IGVAAQNCYKVPKGAFTGEIS-------------PAMI
TIM 0000304*71688---------010000101547*14401110-------------0300
    THHHHHHHHS.TT---------PPPPPP...SSSSBS.SS...-------------HHHH

    HHHHHHHHHHHHHTGGG...S.PPPPSSGGGGTT.SPPPP.......TT..HHHHHHHHH
LDH ***0*4327*26*15**2*09*1220*92540440700002141*8**845925100800
    EDKLKGEMMDLQHGSLFLHTAKIVSGKDYSVSAGSKLVVITAGARQQEGESRLNLVQRNV

    KDIGAA-----------WVILGH--SERRHVFGESDELIGQKVAHALAEGLGVIACIGEK
TIM *71205-----------201000--0203*655246*30070033019550000000109
    HHHT..-----------PPPP..--HHHHHHH...HHHHHHHHHHHHHTT..PPPPPPP.

    HHHHHHHHHHHHH.TT.PPPP..SS----------HHHHHHHHHHHHT..GGGPPE.TT-
LDH 5608*104502*507*000000083----------000002003*42615974000100-
    NIFKFIIPNIVKHSPDCIILVVSNP----------VDVLTYVAWKLSGLPMHRIIGSGC-

    LDEREAGITEKVVFQETKAIADNVKDWSKVVLAYEP---------VWAIGTGKTAT----
TIM 3*85*83528*104*20*102*41*62860000000---------1227955**24----
    HHHHHHTTHHHHHHHHHHHHHHH....TTPPPPPPP---------GGGSSSSS...----

    --------HHHHHHHHHHHHHHHTS.TTTPP..B.BSSSTT..B.GGG.AATTAAHHHHS
LDH --------12004507*300**776*3750606000251*600113220338*86337*9
    --------NLDSARFRYLMGERLGVHSCSCHGWVIGEHGDSVPSVWSGMNVASIKLHPLD

    -----------PQQAQEVHEKLRGWLKTHVSDAVAVQS------------RIIYGGSVTG
TIM -----------3*6029008*03340*9*44*710770------------1001018045
    -----------HHHHHHHHHHHHHHHHHHH.HHHHHHS------------PPPP.S...T

    S..SSSSSSTHHHHHHHHHHHHHHHHHSS..HHHHHHHHHHHHHHHTT..AAAAAAAA.T
LDH 6615***7456039401841**48**85930*310*1005003002*7778610000107
    GTNKDKQDWKKLHKDVVDSAYEVIKLKGYTSWAIGLSVADLAETIMKNLCRVHPVSTMVK

    GNCKELASQHDVDGFLVGGASLKP-----------EFVDIINAKH-------------
TIM 440*70152*400001015207*7-----------50290151**-------------
    THHHHHHTSTT..PPPPSGGGGST-----------HHHHHHT...-------------

    TSSS..SS---.AAAAAAAAATTAAEAA......HHHHHHHHHHHHHHHHH...S...
LDH *5350*54---00000002026*024*35*3*288706*906*007509*12*3****
    DFYGIKDN---VFLSLPCVLNDHGISNIVKMKLKPNEEQQLQKSATTLWDIQKDLKFF
```

Fig. 2. Manually derived alignment of triose phosphate isomerase (TIM) with lactate dehydrogenase based on residue environments. Line 1 (TIM)/3 (LDH): amino acid sequence, Line 2: residue accessibility (0 = 0–9%, 9 = 90–99%, * > 99%), Line 3 (TIM)/1 (LDH): secondary structure (H = $\alpha$-helix, A = antiparallel strand, P = parallel strand, G = 3/10 helix, otherwise coil).

length. A very good example of two topologies which cannot be distinguished after encoding into environmental classes is an $(\alpha\beta)_8$ barrel (a "TIM barrel") and a parallel $\alpha\beta$ sandwich (a Rossmann fold). In this case both topologies comprise alternating $\alpha$ and $\beta$ structure, where the strands are mostly inaccessible to solvent. Providing that the $\alpha\beta$ sandwich is of sufficient size, or if flanking domain regions provide additional secondary structural elements (the Rossmann domain itself typically has only 6 strands), then the 1D descriptors of the two structures are almost identical. This is illustrated in Fig. 2, where the secondary structure and accessibility of TIM (triose phosphate isomerase) has been manually aligned with those of lactate dehydrogenase.

The factor that limits the scope of the search for a stable threading is packing. Whilst the sequence of any isolated helix could substitute for any other, the sequences for a packed pair of helices are much more highly constrained. For a complete protein structure, solvation effects also come into play. In general, then, for a globular protein, the threading of its sequence onto its structure is constrained by local interactions (in the example given, the required formation of a helix), long-range pairwise interactions (helix–helix packing for example) and solvation effects, which are primarily governed by the periodic accessibilities of exposed helices and strands.

In view of this, we should like to match a sequence to a structure by considering the plethora of detailed pairwise interactions, rather than averaging them into a crude environmental class. However, incorporation of such non-local interactions into standard alignment methods such as the algorithm of Needleman and Wunsch [11], has hitherto proved computationally impractical. Possible solutions to this computational problem will be discussed later.

## *8. Formulating a model evaluation function*

The general approach described here employs a set of information theoretic potentials of mean force [25,26]. These potentials associate event probabilities with statistical free energy. If a certain event is observed with probability $p$ (say the occurrence of a leucine residue $\alpha$-carbon and an alanine $\alpha$-carbon at a separation of 5 Å) we can associate an "energy" with this event by the application of the inverse Boltzmann formula:

$$E = -kT\ln(p)\,.$$

The constant $-kT$ may be ignored, in which case the units are no longer those of free energy but of *information* (in units of nats). For simplicity, we have also ignored the additional term $Z$, known as the Boltzmann sum, a clear explanation of why this is acceptable is given by Sippl [26]. The important point about both free energy and information entropy formulations of probability is that the resulting values are additive. Consider two independent events with probabilities $p$ and $q$, respectively. The probability of both events occurring together is simply $pq$, but multiplication is difficult to implement in pattern matching algorithms. Transforming the combined probability $pq$ by taking logs provides the following useful result:

$$\ln(pq) = \ln(p) + \ln(q)\,.$$

Therefore the important part of the calculation of potentials of mean force, and the related techniques of information theory is simply converting probabilities to log-likelihoods.

The real computational key to this approach to transforming probabilities into energy-like parameters is really so that we can transform the problem of multiplying probabilities into the problem of adding related terms. In general it is relatively easy to handle additive terms algorithmically, but very hard to handle values which need to be multiplied. If it makes it easier to follow, readers who are happier working with units of information than units of energy can simply erase the $-kT$ terms from the equations which follow, and even change the base of the logarithms to base-2 so that the scores can be expressed in units of "bits".

Typically we are interested in relative rather than absolute probabilities. Taking the above example, it is of little interest to know how probable it is that a leucine α-carbon and an alanine α-carbon are found to be separated by 5 Å. Of much greater interest is the question of how probable this leucine–alanine separation is in comparison with other residue pairs. If the probability of *any* residue pair having an α-carbon separation of $s$ is $f(s)$ and the frequency of occurrence for residue pair $ab$ is $f_{ab}(s)$ then we can write down the potential of mean force as follows:

$$\Delta E_{ab}(s) = -kT \ln\left[\frac{f_{ab}(s)}{f(s)}\right].$$

Sippl divides this potential into a set of potentials relating to different topological levels $1, \ldots, k$, which is simply the residue pair sequence separation. For the tripeptide sequence MFP, $k=1$ for residue pairs MF and FP, with $k=2$ for residue pair MP. In reality, probability density functions $f_k(s)$ and $f_k^{ab}(s)$ are unknown and must be replaced by the relative frequencies observed in the available structural database denoted $g_k(s)$ and $g_k^{ab}(s)$, respectively, where $s$ is typically divided into 20 intervals for sampling. As there are 400 residue pairs (sequence asymmetry is assumed) and only some 15 000–20 000 residues in the set of non-homologous protein structures, the observed frequency distributions $g_k^{ab}(s)$ are only weak approximations of the true probability densities and must therefore be corrected to allow for the very small sample size. By considering the observation process as the collection of information quanta, Sippl suggests the following transformation:

$$f_k{}^{ab}(s) \approx \frac{1}{1+m\sigma} g_k(s) + \frac{m\sigma}{1+m\sigma} g_k{}^{ab}(s),$$

where $m$ is the number of pairs $ab$ observed at topological level $k$ and $\sigma$ is the weight given to each observation. As $m \rightarrow \infty$ this transformation has the required property that the right and left-hand sides of the equation become equal as $g_k^{ab}(s) \rightarrow f_k^{ab}(s)$. Given the number of residues in the database and the small number of histogram sampling intervals it is assumed that $f_k(s) \approx g_k(s)$. From the previous two equations the following formula may be derived:

$$\Delta E_k{}^{ab} = kT \ln(1+m\sigma) - kT \ln\left[1 + m_{ab}\sigma \frac{g_k{}^{ab}(s)}{g_k(s)}\right].$$

The potentials used in this work are calculated exactly as described by Hendlich et al. [25] where pairwise interatomic potentials are derived from a set of non-homologous proteins. The following interatomic potentials are calculated between the main chain N, O, and side chain Cβ: Cβ → Cβ, Cβ → N, Cβ → O, N → Cβ, N → O, O → Cβ, and O → N. In all, 7 pairwise interactions are considered between each pair of residues $i,j$. By excluding interactions between atoms beyond the Cβ atom in each residue, the potentials are rendered independent of specific side chain conformation. Dummy Cβ atoms were constructed for glycine residues and other residues with missing Cβ atoms.

A possible criticism of the mean force potentials proposed by Sippl is that there exists in the force-field a dependence on protein size. The problem lies in the fact that interactions even as distant as 80 Å are taken into account in the calculation of the potentials, and so consequently, the bulk of data for these large distances is derived from large proteins. This was recognized by Hendlich et al. [25], where it was suggested that the ideal case would be for the potentials to be calculated from proteins of roughly equal size to the protein of interest. Unfortunately, this simple solution is generally impractical. The data set used to generate the mean force potentials is already sparse, even before subdivision into size ranges.

In order to render the mean force potentials less dependent on protein chain length, these long-distance interactions must be replaced by a size independent parameter. The first requirement in replacing these interaction parameters is to determine a suitable dividing line which separates short-distance from long-distance interactions. The next step is then to determine the nature of the information encoded by these interactions. Finally, a suitable size-independent parameter can be sought to replace this information.

Consider two protein atoms separated by a distance $d$. Clearly if $d$ is large there will be no significant physical interaction between these atoms. Conversely, if $d$ is small then we might expect there to be some influence, whether it be a hydrophobic effect, an electrostatic effect or even a covalent interaction. If such an influence exists, then we might also expect there to be some residue preferences for the residues containing these atoms, and consequently we would expect some kind of correlation between the two residue identities. This provides a possible way to determine a cut-off distance for meaningful residue–residue interactions. If the identities of two residues can be considered to be independent variables, then these residues (or more correctly, the residue side chains) will probably not be involved in a significant physical interaction.

To determine the degree of dependency between residues separated by a particular distance, some measure of statistical association is required. The method selected here is based on *statistical entropy*, a common concept in statistical physics and information theory. The entropy of a system with $I$ states, where each state occurs with probability $p_i$, is defined as

$$H(x) = -\sum_{i=1}^{I} p_i \ln p_i.$$

Consider two experiments $x$ and $y$ with $I$ and $J$ possible outcomes respectively, each of which occurs with a probability $p_{i\cdot}$ $(i = 1,\ldots,I)$, and $p_{\cdot j}$ $(j = 1,\ldots,J)$. The entropy $H$ of these systems is defined as

$$H(x) = -\sum_{i=1}^{I} p_{i\cdot} \ln p_{i\cdot}, \qquad H(y) = -\sum_{j=1}^{J} p_{\cdot j} \ln p_{\cdot j}.$$

Entropy in this case is essentially defined as the degree of freedom of choice, or more strictly in this case, the degree of equiprobability. If the outcome probabilities in each experiment are equal then the statistical entropy is maximized, as this represents the maximum freedom of choice. If the probability of one outcome is unity (the others of course being zero) then zero entropy is achieved, corresponding to a lack of choice whatsoever.

If we link both experiments, then we can represent the overall outcomes in the form of a *contingency table*. An example of such a linked pair of experiments is the throwing of a pair of dice, in which case the contingency table would have 6 rows and 6 columns, representing the 6 possible outcomes for each die.

The entropy of the combined experiment is:

$$H(x, y) = -\sum_{i,j} p_{ij} \ln p_{ij}.$$

To determine the statistical association between experiments $x$ and $y$, the entropy of $y$ *given* $x$ and $x$ *given* $y$ may be derived. If a knowledge of the outcome of experiment $x$ allows a wholly accurate prediction of the outcome of experiment $y$, then the entropy of $y$ *given* $x$ must be zero. Conversely, if a knowledge of experiment $x$ is found to be of no benefit whatsoever in the prediction of $y$, then the conditional entropy in this case is maximized.

The entropy of $y$ *given* $x$ is as follows:

$$H(y|x) = \sum_i p_i \sum \frac{p_{ij}}{p_{i\cdot}} \ln\left(\frac{p_{ij}}{p_{i\cdot}}\right) = \sum_{i,j} p_{ij} \ln\left(\frac{p_{ij}}{p_{i\cdot}}\right)$$

and the entropy of $x$ *given* $y$:

$$H(x|y) = \sum_i p_i \sum \frac{p_{ij}}{p_{\cdot j}} \ln\left(\frac{p_{ij}}{p_{\cdot j}}\right) = \sum_{i,j} p_{ij} \ln\left(\frac{p_{ij}}{p_{\cdot j}}\right).$$

Finally a suitable symmetric measure of interdependence (known as the *uncertainty*) between $x$ and $y$ is defined thus:

$$U(x,y) \equiv 2\frac{H(y) + H(x) - H(x,y)}{H(x) + H(y)}.$$

An uncertainty between $x$ and $y$ of zero indicates that the two experimental variables are totally independent $(H(x,y) = H(x) + H(y))$, whereas an uncertainty of one $(H(x) = H(y) = H(x,y))$ indicates that the two variables are totally dependent. One would hope that in the case of the two dice experiment previously described, that $U(x,y)$ would be found to be close to zero for a large number of trials, though gluing the dice together would be a sure way of forcing $U(x,y)$ to unity.

Using the uncertainty measure, it is now possible to evaluate residue correlations in protein structures. In this case, the two experimental variables are the identities of two residues separated by a given distance in a particular structure. Using a set of

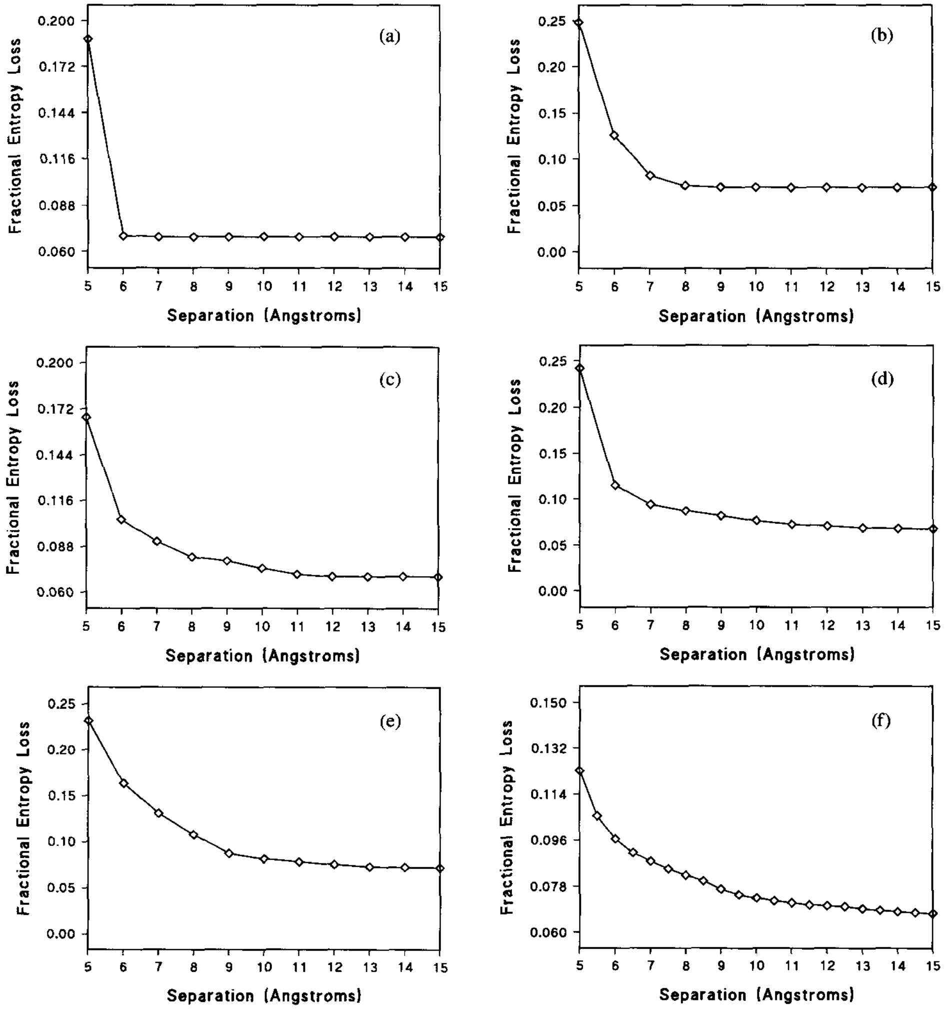

Fig. 3. Uncertainty coefficient (fractional loss of statistical entropy) for residue identities over sequence separations: (a) 1, (b) 2, (c) 3, (d) 4, (e) 5, (f) >10. The maximum observed distances for each sequence separation are as follows: (a) 6.36 Å, (b) 9.72 Å, (c) 13.08 Å, (d) 16.45 Å, (e) 19.43 Å, (f) >32.97 Å. Points beyond these distances have no meaning.

102 chains as listed in Jones et al. [38], six 20×20 contingency tables were set up for each distance range. The first 5 tables were constructed by counting residue pairs with sequence separations of 1 to 5 (short-*range* interactions), the other being constructed by counting all pairs with sequence separations > 10 (long-range). Values in each table were converted to relative frequencies by normalization.

The plots in Fig. 3 clearly show the ranges over which short-range and long-range effects can be detected statistically. As might be expected, the strongest sequence specific

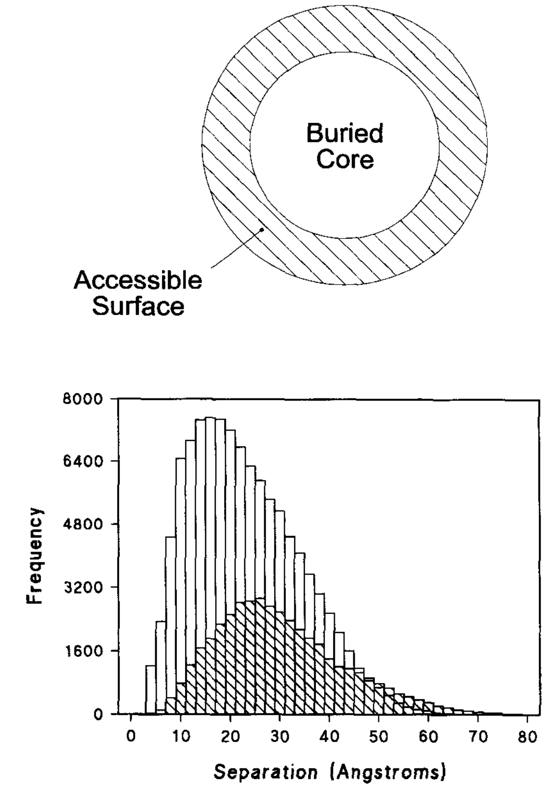

Fig. 4. Distance distributions for accessible (shaded) and inaccessible residue pairs in monomeric protein structures. Buried residues are taken to be those with relative accessibilities <5%; accessible residues with relative accessibilities >50%.

effects are observed across short sequential separations, where steric and covalent effects predominate. Most importantly, both the short- and long-range interactions become undetectable when averaged over distances greater than around 12 Å (though it must be realized that for the very short separations, 1 and 2, it is impossible for the separation to exceed 10 Å). It must be stressed that this does not necessarily imply that the physical forces themselves do not act beyond this distance, only that the effects do not manifest themselves in the selection of amino acid residues.

Bearing in mind the observable extent of detectable sequence specific effects, the calculation of mean force potentials was modified from the method described by Sippl [26]. Rather than taking into account all interactions up to around 80 Å, only atom pairs separated by 10 Å or less were used. However, much useful information remains in the long-distance distributions. Considering a protein molecule as a globule comprising an inner hydrophobic core it is readily apparent that the bulk of the longer pairwise distances will originate from residue pairs distributed on the surface of the globule, which is illustrated in Fig. 4.

As the excluded long-distance potentials clearly only encode information pertaining to the hydrophobic effect, the most logical replacement for these interactions must be a potential based on the solvent accessibility of the amino acid residues in a structure.

## 9. Calculation of potentials

For the short-range potentials, minimum and maximum pairwise distances were determined for each type of atomic pairing at each topological level from a small set of very highly resolved crystal structures. These distance ranges were subdivided into 20 intervals. For the medium and long-range potentials, interactions were sampled over the range 0–10 Å with a fixed sampling interval of 2 Å. A small selection of the pairwise interaction potentials is shown in Fig. 5.

As discussed in the previous section, in addition to the pairwise potentials (and in place of the long-range, long-distance interactions), a solvation potential was also incorporated.

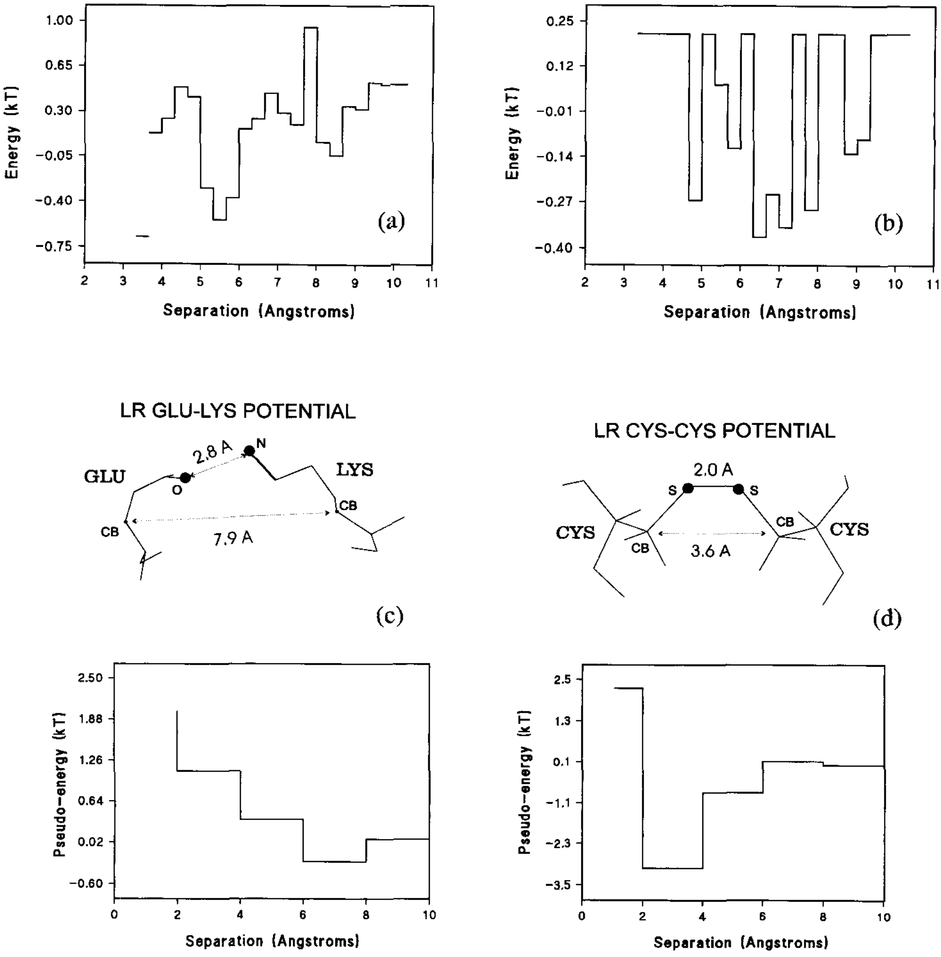

Fig. 5. Sample pairwise potentials: (a) Short-range ($k=3$) Ala–Ala Cβ–Cβ, (b) Short-range ($k=3$) Phe–Tyr Cβ–Cβ, (c) Long-range ($k>30$) Glu–Lys Cβ–Cβ, shown below a diagram of the typical geometry for a Glu–Lys salt-bridge, (d) Long-range ($k>30$) Cys–Cys Cβ–Cβ, shown below a diagram of the typical geometry for a disulphide bond.

This potential simply measures the frequency with which each amino acid species is found with a certain degree of solvation, approximated by the residue solvent accessible surface area. The solvation potential for amino acid residue $a$ is defined as follows:

$$\Delta E^{a}_{\text{solv.}}(r) = -kT \ln\left[\frac{f^{a}(r)}{f(r)}\right],$$

where $r$ is the % residue accessibility (relative to residue accessibility in GGXGG extended pentapeptide). Residue accessibilities were calculated using the DSSP program of Kabsch and Sander [39]. The solvation potentials were generated with a histogram sampling interval of 5%. To ensure that subunit or domain interactions did not affect the results, only monomeric proteins were used in the calculation. These solvation potentials clearly show the hydropathic nature of the amino acids and prove to be a more sensitive measure of the likelihood of finding a particular amino acid with a given relative solvent accessibility than the long-distance interaction potentials they are designed to replace.

## *10. Searching for the optimal threading*

Given an efficient means for the evaluation of a hypothetical sequence threading relationship, the problem of finding the optimal threading must be considered. For a protein sequence of length $L$ and a template structure of which $M$ residues are in regular secondary structures, the total number of possible threadings is given by

$$\binom{L}{M} \equiv \frac{L!}{(L-M)!\,M!}.$$

The scale of the search problem for locating the optimal threading of a sequence on a structure amongst all possible threadings may be appreciated by considering bovine pancreatic trypsin inhibitor (Brookhaven code 5PTI) as an example. Of the 58 residues of 5PTI, 30 are found to be in regular secondary structure. Using the above expression the total number of threadings for 5PTI is calculated as $2.9\times10^{16}$. Given that 5PTI is a very small protein, it is clear that the search for optimal threading is a non-trivial problem.

One way to reduce the scale of the problem is to restrict insertions and deletions (indels) to the loop regions. By excluding indels from secondary structural elements, the problem reduces to a search for the optimal set of loop lengths for a protein. Under these conditions threading a sequence on a structure may be visualized as the sliding of beads on an abacus wire, where the beads are the secondary structures, and the exposed wire the remaining loop regions. The restricted threading of a small four helix bundle protein (myohemerythrin) is depicted in Fig. 6. Restricting indels to loop regions in this way reduces the search space in this case from $1.3\times10^{33}$ to just 44 100.

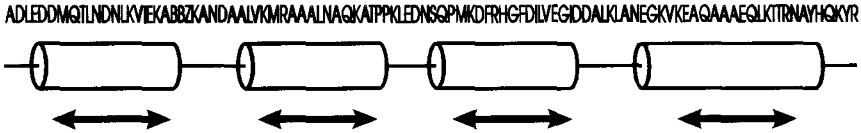
ADLEDDMQTLNDNLKVIEKABBZKANDAALVKMRAAALNAQKATPPKLEDNSQPMKDFRHGFDILVEGIDDALKLANEGKVKEAQAAAEQLKTTRNAYHQKYR

Fig. 6. Diagram illustrating the exhaustive threading procedure. Secondary structures are effectively moved along the sequence like abacus beads. On average, loop lengths vary between 2 and 12.

Unfortunately, even with this extreme restriction on the threading patterns, the search space again becomes unreasonably large for proteins of even average size. The exact number of threadings with secondary structure indels disallowed is a complex function of both sequence length and the number and lengths of constituent secondary structural elements. As a rule of thumb, however, it is found that for $N$ secondary structural elements, with loops of average length (say between 2 and 12 residues), the number of threadings is $O(10^N)$. For typical proteins comprising 10–20 secondary structures, it is clearly not possible to locate the optimal threading by an exhaustive search in a reasonable period of time.

## *11. Methods for combinatorial optimization*

Various means have been investigated for locating the optimal threading of a sequence on a structure. The methods are briefly detailed below.

### *11.1. Exhaustive search*

As demonstrated earlier, for small proteins of $<6$ secondary structural elements, and disallowing secondary structure indels, it is practical to simply search through all possible threadings in order to locate the threading of lowest energy. Unfortunately, the evaluation function is tailored towards average sized globular proteins with hydrophobic cores, whereas the small proteins tend to be less globular and typically lack a hydrophobic core.

### *11.2. Monte Carlo methods*

Monte Carlo methods have been often exploited for conformation calculations on proteins. Two *directed* search procedures have been used in my laboratory: *simulated annealing*, and *genetic algorithms*. Simulated annealing has been exploited in the alignment of protein structures [40], and in the optimization of side chain packing in protein structures [41]. Simulated annealing is a simple random search process. In this instance, random threadings are generated and evaluated using the evaluation function described earlier. Where a proposed threading has a lower energy than the current threading, the proposed threading is accepted. In the case where a proposed threading has a higher energy than the current, it is accepted with probability $p$, where

$$p = \exp\left(-\frac{\Delta E}{kT}\right),$$

and where $\Delta E$ is the difference between the current and the proposed threading energy and $T$ is the current annealing "temperature". After a predefined number of accepted changes, the temperature is slightly reduced. This whole procedure is repeated until no further reduction in threading energy is achieved, at which point the system is said to be frozen. The schedule of cooling is critical to the success of simulated annealing.

Genetic algorithms [42] are similar in concept to simulated annealing, though their model of operation is different. Whereas simulated annealing is loosely based on the principles of statistical mechanics, genetic algorithms are based on the principles of natural selection. The variables to be optimized are encoded as a string of binary digits, and a *population* of random strings is created. This population is then subjected to the genetic operators of selection, mutation and crossover. The probability of a string surviving from one generation to the next relates to its fitness. In this case, low energy threadings are deemed to be fitter than those with higher energies. Each string may be randomly changed in two ways. The mutation operator simply selects and changes a random bit in the string. An alternative means for generating new strings is the crossover operator. Here a randomly selected portion of one string is exchanged with a similar portion from another member of the string population. The crossover operator gives genetic search the ability to combine moderately good solutions so that "super-individuals" may be created.

In use, these methods prove to be capable of locating the optimal threading, but with no guarantee that they will do so in any given run of the threading program. Ideally the results from many runs should be pooled and the best result extracted, which is of course time consuming. A further problem is that the control parameters (the cooling schedule in the case of simulated annealing and the selection, mutation and crossover probabilities in the case of genetic search) need adjustment to match each threading problem individually. Parameters found suitable for threading a protein with 10 secondary structures will generally not be suitable for threading a protein with 20 secondary structures for example. The methods are typically plagued by "unreliability", yet are found to be highly robust. Given a sufficiently slow cooling rate in the case of simulated annealing, or a sufficiently large population of strings in the case of genetic algorithms, and in both cases a sufficient number of runs, very low energy threadings will be found providing they exist at all in the given search space.

### *11.3. Dynamic programming*

It should be apparent that there exists a clear similarity between optimizing the threading of a sequence on a structural template and finding the optimal alignment of two sequences. In such terms, threading is simply the alignment of an amino acid sequence against a sequence of positions in space. At first sight it might well appear that the same dynamic programming methods used in sequence alignment (e.g. ref. [11]) could easily be applied to the threading problem. Unfortunately, this is not the case. In a typical alignment algorithm a score matrix is constructed according to the following recurrence formula:

$$S_{ij} = D_{ij} + \max \begin{cases} & S_{i+1,j+1}; \\ \max\limits_{k=i+2\rightarrow N_{\mathrm{A}}} & S_{k,j+1} - g; \\ \max\limits_{l=j+2\rightarrow N_{\mathrm{B}}} & S_{i+1,l} - g; \end{cases}$$

where $S_{ij}$ is an element of the score matrix, $D_{ij}$ is a measure of similarity between residues $i$ and $j$ in sequences of length $N_A$ and $N_B$, respectively, and $g$ is a gap penalty which may be either a constant or a function of, for example, gap length. By tracing the highest scoring path through the finished matrix the mathematically optimum alignment between the two sequences may be found for the given scoring scheme. In the special case where $D_{ij}$ is a function only of elements $i$ and $j$, dynamic programming alignment algorithms have execution times proportional to the product of the sequence lengths. However, if $D_{ij}$ is defined in terms of non-local sequence positions in addition to $i$ and $j$, dynamic programming no longer offers any advantage; the alignment effectively requires a full combinatorial search of all possible pairings. In the case of the evaluation function defined here, in order to determine the energy for a particular residue, all pairwise interactions between the residue in question and every other residue in the protein need to be calculated. In other words, in order to evaluate the threading potentials in order to fix the location of a single residue, the location of every other residue needs to have been fixed beforehand.

By excluding the medium and long-range ($k > 10$) pairwise terms from the evaluation function and by considering only interactions between residues in the same secondary structural element, dynamic programming can be applied to the problem. For example, consider a case where a template comprising a single 10 residue helical segment is being matched against a 100 residue sequence. Discounting the possibility of indels in the helix itself, there are 91 possible alignments between the helical template and the sequence. As indels may not occur in the helix, for any given position in the sequence ($i = 1,...,91$), all possible interhelical pairwise interactions are defined. However, this simplification allows only local conformational effects to be considered. Packing between secondary structures may be evaluated only by means of the solvation potentials and not by any pairwise terms. Clearly it would be ideal to devise an efficient dynamic programming method capable of taking non-local pairwise terms into account.

### *11.4. Double dynamic programming*

The requirement here to match pairwise interactions relates to the requirement of structural comparison methods. The *potential environment* of a residue $i$ is defined here as being the sum of all pairwise potential terms involving $i$ and all other residues $j \neq i$. This is a similar definition to that of a residue's *structural environment*, as described by Taylor and Orengo [43]. In the simplest case, a residue's structural environment is defined as being the set of all inter-Cα distances between residue $i$ and all other residues $j \neq i$. Taylor and Orengo propose a novel dynamic programming algorithm (known as double dynamic programming) for the comparison of residue structural environments, and it is a derivative of this method that I have used for the effective comparison of residue potential environments.

Let $T_m$ ($m = 1, \ldots, M$) be the elements of a structural template, and $S_n$ ($n = 1, \ldots, N$) be the residues in the sequence to be optimally fitted to the template. We wish to determine a score $Q(T_m, S_n)$ for the location of residue $n$ at template position $m$. In order to achieve this, the optimal interaction between residue $n$ and all residues $q \neq n$ conditional on the matching of $T_m$ *and* $S_n$ is calculated by the application of the standard dynamic programming algorithm. We define two matrices: a low-level matrix $\boldsymbol{L}$ (more precisely a *set* of low-level matrices), and a high-level matrix $\boldsymbol{H}$ into which the best paths through each of the low-level matrices are accumulated. The calculation of a single low-level matrix $\boldsymbol{L}$ is illustrated in Fig. 7.

For each $m, n$, the total potential of mean force, $Z(m,n,p,q)$, may be calculated for each $p, q$ where $p$ is again an element in the structural template, and $q$ a residue in the object sequence:

$$Z(m,n,p,q) = \Delta E^{S_q}_{\text{solv.}}(A_p) + \begin{cases} \Delta E^{S_nS_q}_{(q-n)}(d_{mp}), & q > n,\ p > m; \\ \Delta E^{S_qS_n}_{(n-q)}(d_{pm}), & q < n,\ p < m; \\ 0, & q = n,\ p = m \quad \text{or} \quad q = n, p \neq m; \\ U, & q < n,\ p > m \quad \text{or} \quad q > n,\ p < m; \end{cases}$$

here $U$ is a large positive constant penalty which forces the final path to incorporate pair $m, n$, $A_p$ is the accessibility of template position $p$, $d_{mp}$ and $d_{pm}$ are elements of the template interatomic distance matrix and the pairwise, $\Delta E^a_k b(r)$, and solvation, $\Delta E^a_{\text{solv}}(s)$, terms are as defined previously. Pairwise terms are summed over all required atom pairs (Cβ → Cβ, Cβ → N for example), using appropriate values from the distance matrix, though typically, for computational efficiency, the low-level matrices are calculated using the Cβ → Cβ potential alone.

The low-level matrix $\boldsymbol{L}$ is then calculated using the standard NW algorithm:

$$L_{pq} = Z(m,n,p,q) + \min \begin{cases} & L_{p+1,q+1}; \\ \min\limits_{r=p+2 \to N_{\text{A}}} & L_{r,q+1} + g(S_p); \\ \min\limits_{s=q+2 \to N_{\text{B}}} & L_{p+1,s} + g(S_p); \end{cases}$$

where $S_p$ is the secondary structural class (helix, strand, coil) of template position $p$. $g(S_p)$ is a simple secondary structure dependent gap penalty function:

$$g(S_p) = \begin{cases} G_{\text{s}}, & S_p = \text{helix, strand,} \\ G_{\text{c}}, & S_p = \text{coil,} \end{cases}$$

where $G_s$ and $G_c$ are both positive constants, and $G_s \gg G_c$. As with the application of the algorithm to structure comparison, it is found to be advantageous to accumulate the low-level matrix scores along each suggested low-level path, and so the paths from each low-level matching, conditional on each proposed match between $T$ and $S$, for which the path scores exceed a preset cut-off, are accumulated into $\boldsymbol{H}$ thus:

$$H'_{pq} = H_{pq} + \min \begin{cases} & L_{p+1,q+1}; \\ \min\limits_{r=p+2 \to N_{\text{A}}} & L_{r,q+1}; \\ \min\limits_{s=q+2 \to N_{\text{B}}} & L_{p+1,s}; \end{cases}$$

for all $p, q$ along the optimum traceback path in $\boldsymbol{L}$.

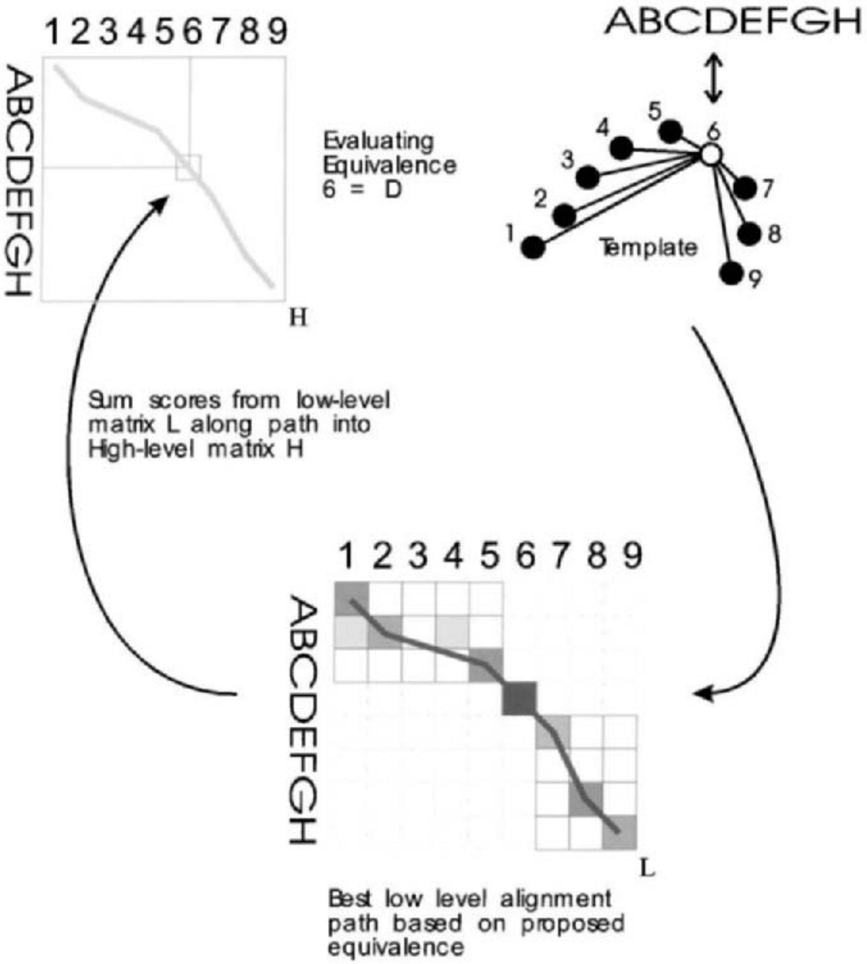
1 2 3 4 5 6 7 8 9
ABCDEFGH
H
Evaluating Equivalence 6 = D
ABCDEFGH
1
2
3
4
5
6
7
8
9
Template
Sum scores from low-level matrix L along path into High-level matrix H
1 2 3 4 5 6 7 8 9
ABCDEFGH
L
Best low level alignment path based on proposed equivalence

Fig. 7. Calculation of a single low-level matrix path as part of the double dynamic programming algorithm.

The overall operation may be thought of as the matching of a distance matrix calculated from the template coordinates with a *probability* matrix (in practice, an information matrix) calculated from the object sequence.

The final alignment (matrix $\boldsymbol{F}$) is generated by finding the best path through the final high-level matrix thus:

$$F_{pq} = H_{pq} + \min \begin{cases} & F_{p+1,q+1}; \\ \min\limits_{r=p+2\to N_{\mathrm{A}}} & F_{r,q+1} + g(S_p); \\ \min\limits_{s=q+2\to N_{\mathrm{B}}} & F_{p+1,s} + g(S_p). \end{cases}$$

In the above expressions, each instance of the NW algorithm has been formulated in terms of *minimizing* a cost function, as the scores in this application are energies. In practice, however, these expressions could be converted trivially to a form where a score is maximized, simply by changing the sign of the calculated energy values, or by just leaving the interaction propensities in units of information. Where mention is made of a *high-scoring* path (which is rather more familiar terminology in sequence comparison) in the following pages, then this should be taken as referring to a path with *low* energy.

As described, the double dynamic programming algorithm is too slow to be useful for fold recognition. The efficient algorithm by Gotoh [44] for calculating the NW score matrix is $O(MN)$ where MN is the product of the two sequence lengths. Double dynamic programming involves the use of this algorithm for all $MN$ possible equivalent pairs of residues, giving an overall algorithmic complexity of $O(M^2N^2)$. On a typical present day workstation, a single instance of the Gotoh algorithm for two sequences of length 100 can be performed in around 0.1 CPU s. Multiplying this time by $100^2$ provides an estimate of 1000 CPU s to complete a single double dynamic programming comparison, which is clearly too slow to be applied to the fold recognition problem, where many hundreds of instances of the double dynamic programming algorithm would be required. Furthermore, many comparisons would involve sequences or structures longer than 100 residues in length. The absurdity of the problem becomes apparent when it is realized that to compare a single sequence 500 residues in length with a structure of similar size, roughly 12 CPU h would be required. Even for a fold library with a few hundred folds, a single search would take several months to complete.

Clearly, if the double dynamic programming algorithm is to be of use, short-cuts must be taken. The most straightforward short-cut to take is to apply a window to both levels of the algorithm. This is very helpful, but still insufficient to make the algorithm convenient for general use. Orengo and Taylor proposed the use of a prefiltering (*residue selection*) pass to exclude unlikely equivalences before going on to calculate a low-level path. In the case of structural comparison, Orengo and Taylor initially select pairs of residues from the two structures primarily on the basis of similarities in their relative solvent accessibility and main chain torsion angles. Residue pairs found to be in different local conformations, and with differing degrees of burial are clearly unlikely to be equivalenced in the final structural alignment, and consequently should be excluded as early as possible in the double dynamic programming process. Unfortunately for sequence–structure alignment, it is not possible to select residue pairs on real measured quantities such as accessibility or torsion angles. However, these quantities could in principle be *predicted* for the sequence under consideration, and these predicted values then compared with the real values observed in the template structure. In practice, these values are not actually predicted directly, but the method proposed here certainly makes use of the same principles that might be employed in their prediction.

## *12. Residue selection for sequence–structure alignments*

The residue selection stage of optimal sequence threading involves the summation of local interaction potential terms $\Delta E_k^a b$ over all residue pairs in overlapping windows of length $L$, where $L$ is a small constant odd-number over the range, say, 5–31. Similarly the solvation terms are summed for each of the $L$ residues. The window is clipped appropriately if it spans either the N- or C-terminus of either the sequence or the structure, for example the window length for the first and last residue in either cases would be $\frac{1}{2}(L+1)$. To equalize the contribution of the pairwise terms and the solvation terms, the average energy is calculated and summed for both, giving a total energy for the sequence–structure fragment of:

$$S_{mn} = E(\text{fragment}) = \frac{2\sum_{i=1}^{L-1}\sum_{j=i+1}^{L} E_{\text{pair}}^{ij}}{L(L-1)} + \frac{\sum_{k=1}^{L} E_{\text{solv}}^{k}}{L}.$$

Energies are calculated for every sequence fragment threaded onto every structure fragment, and the results stored in a selection matrix $S$. Using the residue selection step, the number of initial pairs $(m,n)$ for which low-level paths need to be calculated is reduced up to 100-fold.

## *13. Evaluating the method*

In the original paper describing the idea of optimal sequence threading [38] the method was applied to 10 examples with good results, in particular it proved able to detect the similarity between the globins and the phycocyanins, and was the first sequence analysis method to achieve this. Also of particular note were the results for some $(\alpha\beta)_8$ (TIM) barrel enzymes and also the b-trefoil folds: trypsin inhibitor DE-3 and interleukin 1β. The degree of sequence similarity between different $(\alpha\beta)_8$ barrel enzyme families and between trypsin inhibitor DE-3 and interleukin 1β is extremely low (5–10%), and as a consequence of this, sequence comparison methods had not proven able to detect these folds.

Although these results are good, it is fair to argue that in all cases the correct answers were already known and so it is not clear how well they would perform in real situations where the answers are not known at the time the predictions are made. The results of a very ambitious world wide experiment were published in a special issue of the journal "PROTEINS", where an attempt was made to find out how successful different prediction methods were when rigorously blind-tested (i.e., applied to problems where the answer was not known at the time). In 1994, John Moult and colleagues approached X-ray crystallographers and NMR spectroscopists around the world and asked them to deposit the sequences for any structures they were close to solving in a database. Before these structures were made public, various teams around the world were then challenged with the task of predicting each structure. The results of this experiment were announced at a meeting held at Asilomar in California, and this ambitious experiment has now become widely known as the Asilomar Experiment (or more commonly the Asilomar Competition).

The results for the comparative modelling and *ab initio* sections offered few surprises, in that the *ab initio* methods were reasonably successful in predicting secondary structure but not tertiary, and homology modelling worked well when the proteins concerned had very high sequence similarity. The results for the fold recognition section [45], however, showed great promise. Overall, roughly half of the structures in this part of the competition were found to have previously observed folds. Almost all of these structures were correctly predicted by at least one of the teams. The threading method described in this chapter proved to be the most successful method [46], with 5 out of 9 folds correctly identified, and with a looser definition of structural similarity, 8 out of 11 correct.

## *14. Software availability*

A program, called THREADER, which implements the threading approach to protein fold recognition described here has been made widely available to the academic community free of charge, and can be downloaded over the Internet from the URL
`http://globin.bio.warwick.ac.uk/~jones/threader.html`

## *References*

[1] Levinthal, C. (1968) Chim. Phys. 65, 44–45.
[2] Rooman, M.J., Kocher, J.P.A. and Wodak, S.J. (1991) J. Mol. Biol. 221, 961–979.
[3] Moult, J. and Unger, R. (1991) Biochemistry 30, 3816–3824.
[4] Chan, H.S. and Dill, K.A. (1990) Proc. Natl. Acad. Sci. USA 87, 6388–6392.
[5] Gregoret, L.M. and Cohen, F.E. (1990) J. Mol. Biol. 211, 959–974.
[6] Doolittle, R.F. (1992) Prot. Sci. 1, 191–200.
[7] Orengo, C.A., Jones, D.T. and Thornton, J.M. (1994) Nature 372, 631–634.
[8] Chothia, C. (1992) Nature 357, 543–544.
[9] Cohen, F.E., Sternberg, M.J.E. and Taylor, W.R. (1982) J. Mol. Biol. 156, 821–862.
[10] Taylor, W.R. (1991) Prot. Eng. 4, 853–870.
[11] Needleman, S.B. and Wunsch, C.D. (1970) J. Mol. Biol. 48, 443–453.
[12] Taylor, W.R. (1986a) J. Mol. Biol. 188, 233–258.
[13] Gribskov, M., Lüthy, R. and Eisenberg, D. (1990) Methods Enzymol. 188, 146–159.
[14] Bashford, D., Chothia, C. and Lesk, A.M. (1987) J. Mol. Biol. 196, 199–216.
[15] Barton, G.J. (1990) Methods Enzymol. 188, 403–428.
[16] Pearl, L.H. and Taylor, W.R. (1987) Nature 328, 351–354.
[17] Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer, E.F., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T. and Tasumi, M. (1977) J. Mol. Biol. 112, 535–542.
[18] Novotny, J., Bruccoleri, R.E. and Karplus, M. (1984) J. Mol. Biol. 177, 787–818.
[19] Brooks, B., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S. and Karplus, M. (1983) J. Comput. Chem. 4, 187–217.
[20] Novotny, J., Rashin, A.A. and Bruccoleri, R.E. (1988) Proteins 4, 19–30.
[21] Eisenberg, D. and McLachlan, A.D. (1986) Nature 319, 199–203.
[22] Baumann, G., Frommel, C. and Sander, C. (1989) Prot. Eng. 2, 329–334.
[23] Holm, L. and Sander, C. (1992) J. Mol. Biol. 225, 93–105.
[24] Lüthy, R., Bowie, J.U. and Eisenberg, D. (1992) Nature 356, 83–85.
[25] Hendlich, M., Lackner, P., Weitckus, S., Floeckner, H., Froschauer, R., Gottsbacher, K., Casari, G. and Sippl, M.J. (1990) J. Mol. Biol. 216, 167–180.
[26] Sippl, M.J. (1990) J. Mol. Biol. 213, 859–883.
[27] Casari, G., Sippl, M.J. (1992) J. Mol. Biol. 224, 725-732.
[28] Crippen, G.M. (1991) Biochemistry 30, 4232–4237.

[29] Maiorov, V.N. and Crippen, G.M. (1992) J. Mol. Biol. 227, 876–888.
[30] Ponder, J.W. and Richards, F.M. (1987) J. Mol. Biol. 193, 775–791.
[31] Bowie, J.U., Clarke, N.D., Pabo, C.O. and Sauer, R.T. (1990) Proteins 7, 257–264.
[32] Orengo, C.A., Brown, N.P. and Taylor, W.R. (1992) Proteins 14, 139–167.
[33] Bowie, J.U., Lüthy, R. and Eisenberg, D. (1991) Science 253, 164–170.
[34] Bork, P., Sander, C. and Valencia, A. (1992) Proc. Natl. Acad. Sci. USA 89, 7290–7294.
[35] Finkelstein, A.V. and Reva, B.A. (1991) Nature 351, 497–499.
[36] Chou, P.Y. and Fasman, G.D. (1974) Biochemistry 13, 212–245.
[37] Fauchere, J.L. and Pliska, V.E. (1983) Eur. J. Med. Chem. 18, 369–375.
[38] Jones, D.T., Taylor, W.R. and Thornton, J.M. (1992) Nature 358, 86–89.
[39] Kabsch, W. and Sander. C. (1983) Biopolymers 22, 2577–2637.
[40] Šali, A. and Blundell, T.L. (1990) J. Mol. Biol. 212, 403–428.
[41] Lee, C. and Subbiah, S. (1991) J. Mol. Biol. 217, 373–388.
[42] Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, Reading, MA.
[43] Taylor, W.R. and Orengo, C.A. (1989) J. Mol. Biol. 208, 1–22.
[44] Gotoh, O. (1982) J. Mol. Biol. 162, 705–708.
[45] Lemer, C.M.R., Rooman, M.J. and Wodak, S.J. (1995) Proteins 23, 337–355.
[46] Jones, D.T., Miller, R.T. and Thornton, J.M. (1995) Proteins 23, 387–397.