Title: **Protein Threading**

Manuscript version: 15 Nov. 03
For submission to "The Proteomics Handbook"

Andrew E. Torda
Zentrum Für Bioinformatik
University of Hamburg
D-20146 Hamburg
Germany
phone: +49-40-42838 7331
fax: +49-40-42838 7332
torda@zbh.uni-hamburg.de

keywords: "protein structure prediction", "sequence alignment", "force field", "sequence to structure alignment", "alignment quality"

Abstract

Given a protein's sequence, one may try to predict its structure by reference to basic physics or even by searching against some more pragmatic quasi-energy or score function. Threading attempts to solve what should be a simpler problem – looking through the set of currently known structures and identifying the ones which are most likely to be appropriate for the sequence of interest. Unlike pure sequence-based methods, the calculations should use known structural information. It remains to be seen if threading will be obsolesced by the best sequence based methods or the newest approaches which do not even use a template structure.

1. Introduction

Theoreticians have been trying to predict protein structure based on sequence information for decades. Literally, more than a quarter century ago, there were optimistic reports that one could use simulation methods to calculate the structure of a small protein given only its sequence.(1,2) To this day, devotees of this approach persevere and may ultimately win over the problems with force fields and the enormous search space. In the meantime, a class of protein structure methods have developed travelling under names such as protein threading and fold recognition.

In the most general case, protein structure prediction is a truly ferocious problem whose size can be made clear by a model calculation. Imagine that every peptide plane (ω) angle is fixed, planar and *trans*. At every residue, one still has the phi and psi (φ, ψ) backbone angles and there might be two or three preferred local conformations. Even in this unrealistically simple case, a protein of 100 residues has between 10^{30} (2^{100}) and 10^{47} (3^{100}) conformations to be considered. These numbers come without even considering sidechains or the fact that backbone conformations are continuous variables and do not fall into two or three discrete locations. At the risk of being a doomsayer, one could also note that computers double their speed every few years, but the size of the computational problem doubles with every extra amino acid. If you can predict the structure of a 50 residue protein this year, it could be a few more years until you can do 51 residues.

Rather than give up, one can look for a simpler version of the problem or a subset which might be solvable. Proteins probably do not manage to fold into every shape a polymer chemist could imagine(3). Instead, there may only be a finite number of protein folds in nature(4-12) and certain kinds of structure seem to be remarkably popular amongst apparently unrelated sequences.(13-20) This has an important consequence. Even when an experimentalist may not expect any similarity, the structure they are about to solve may be quite similar to one that is already known. In recent years, less than 15 % of structures deposited in the protein data bank (21) could even be considered new folds. This is the rationale for the entire area of threading or fold recognition. The protein sequence of interest may have no detectable sequence homology to anything of known structure, but there may well be some similar structure waiting to be recognised. If one can recognise the related structure and do a sequence to structure alignment calculation, one should produce a useful model. It would be even better if one could detect those cases where the method will fail and the sequence will fold to a new structure. Unlike normal sequence comparison, the alignment method should take advantage of the structural information of each template.

Some of the statements above are poorly quantified and no two groups may agree on what constitutes a new fold and how often one is found. At the same time, one can quote some findings on when sequence similarity is sufficient to infer similarity. It is often said that if a sequence has more than 30 % identity to a known structure, it is possible to build a reasonable model, but at 20 to 25 % the similarity may be purely coincidental. In practice, this rule of thumb should not be used. For example, Brenner *et al* give the example of a pair of proteins with 39 % sequence identity, but no detectable structural similarity.(22) Rather than look for a single number (sequence identity), one must look at the length of the proteins and aligned regions. Intuitively, it is obvious that 40 % similarity over a small peptide is much more likely to happen by chance than 40 % over 500 residues. This issue has been addressed by comparing large numbers of protein pairs.(22,23) Basically, for 50 residues, you would want 40 % sequence identity before deeming it reliable, but for 250 residues, 25 % might suffice. These numbers are purely statistical, so there is always the distinct possibility that a weak sequence identity does not reflect structural similarity. A better approach is to resist the temptation to concentrate on pairwise numbers. Sequence database searching programs such as FASTA (24) and BLAST(25) estimate the reliability of a sequence match by looking at it in the context of the whole library of sequence scores. More recent, iterated versions of BLAST(26), render the interpretation of pairwise sequence identity even more meaningless. Because programs such as PSI-BLAST work with a sequence profile, a database hit is often statistically reliable, even with less than 20 % sequence identity.

With these results in mind, one is left with an unsatisfying, but practical way to decide whether or not a threading calculation is of interest. If a simple database search finds a reliable homologue of known structure for a sequence, it is the best way to build a model. If a careful, exhaustive, iterated database search cannot find a statistically reliable homologue or if you wish for a confirmation of your beliefs, a threading calculation is called for.

2. Threading Overview

For a threading calculation, there are some elements common to most programs. Firstly, you have a sequence of interest and a library of templates or known structures as shown in Fig 1. Presumably, these are protein data bank structures and the library contains all known protein folds. Next, one takes the sequence and "threads" it through each template in the library as shown in Fig. 2 . The word threading implies that one drags the sequence

(ACDEFG...) step by step through each location on each template, but really one is searching for the best arrangement of the sequence as measured by some score or quasi-energy function. In the third alignment in Fig. 2, the sequence of interest has been aligned so it skips over part of the template. Finding the best arrangement of residues, including these gaps and insertions is the problem of sequence to structure alignment, discussed below. Finally, all the candidate models with their scores are collected in Fig. 3. The best scoring (lowest energy) one is then taken as the structure prediction.

Before considering technical details, this simple picture highlights some problems. Firstly, the result will depend on the size and details of the library at the first step. Typically, different groups will have libraries ranging from 500 to 5000 members and there is definitely no consensus as to the optimal size. On the one side, the library should be small. Threading calculations are often slow, so one may want to use the smallest possible library. At the same time, threading score functions are far from perfect. The closer a template is to the correct answer, the more likely the sequence is to score well on it. Thus libraries should be large. However, imagine you include 10 small variations on one particular protein fold, but one representative from another. Statistically, the well-represented fold is more likely to score well by chance. Thus, libraries should be small. Finally, there is no agreed way to select the particular members.

One could argue that library members should be single chains or domains. One could argue that from a protein family, one should select the member that has the best quality coordinates or the one that is in some way most representative of the family of structures. Continuing in this vein, the idea of representative structures implies that one has already clustered all known proteins down to a set of families. This could be based on sequence identity or some measure of structural similarity. Finally, one is not even limited to simple PDB coordinates. Madej *et al* used a library based on extracted cores from proteins (27) and some have suggested that the structures in the library could be optimised so as to make the ranking of models statistically more reliable.(28)

The simple set of pictures also introduce the next questions. One needs some way to score the sequences and structures and then one will need a way to find the best alignment of a sequence on a template structure.

3. Score functions

Much of computational chemistry is centred about finding the best conformation for some molecule. In protein calculations, this usually involves a classical atomistic model for the potential energy of a system.(29-32) In the case of protein threading, one is not bound by this philosophy. One really just wants a score function which is capable of recognising correct arrangements of protein residues. It need not perform all the feats of a conventional force field or work in the same application areas. For example, in the procedure described so far, one need never take the derivative of score with respect to coordinates, although this would usually be an essential step in many energy minimisation schemes or dynamics simulations. Similarly, if a score function is only used in a threading context, it will never be faced with atoms hitting each other, stretched bonds, distorted angles or any of the other situations which might confront a normal force field. This should mean that it is easier to build a through-space, threading score function than a full force field for molecular mechanics calculations.

Threading score functions are also usually more coarse-grained than those used in a real energy calculation. In a threading calculation, the sequence residues are placed on the backbone of the template structure and from there, one can calculate ideal coordinates for the C^{β} atom. One does not know where the rest of the residue is so it will be extremely difficult to use a score function which uses the coordinates of all atoms. Consequently, a threading score function usually represents each residue by one or a few interaction sites. Often, most of the chemical identity of a residue comes from an interaction site located at the C^{β} residue or a point closer to the sidechain centre of mass.

With this level of representation, it is not common to rely on pure physics. For example, a threading score function does not usually have a term like Coulombs law for electrostatics or a Lennard-Jones term for other atomic interactions. Instead, there are two common approaches to building a score function: (i) potentials of mean force and (ii) from an optimisation calculation.

Potentials of mean force are described in statistical mechanics textbooks, often based on the distributions of particles in simulations.(33) At a coarse-grained level, suitable for threading, there were parameterised in the 1970's, 1980's (34,35) and repeatedly since.(36,37) In the protein literature, they usually travel under names such as statistics- or Boltzmann-based force fields or score functions and sometimes even knowledge-based force fields. The principle is easily illustrated by example. If we know the concentration of two species (particle types A and B), we can calculate how often they will be observed at a certain distance from each other by chance. If AB pairs are seen more often than expected at 5 Å, the system is behaving as if there is some kind of energy minimum between the particles at 5 Å. If the pairs of particles are seen less often than expected at that distance, it appears that the interaction is unfavourable at that distance. To formalise this, one must remember the words of Herr Boltzmann and refer to some free energy, G which is a function of the distance r_{AB} between particles of types A and B:

$$G(r_{AB}) = kT \ln \frac{\rho_{r_{AB}}}{\rho^0_{r_{AB}}}$$

k and T have their normal meanings of Boltzmanns constant and temperature. $\rho_{r_{AB}}$ is the observed frequency of AB pairs at distance r. $\rho^0_{r_{AB}}$ is less obvious. It is the frequency of AB pairs at distance r you would expect to see by chance. This formulation is very general and is easy to apply to proteins. Instead of considering particles A and B, you might consider C^β atoms on Ala and Trp residues. Then you could build a potential of mean force for Ala/Trp C^β atoms and you could do this for every combination of amino acids. One could even parameterise this kind of function in terms of torsion angles or any other property that seems to be important for determining a protein's structure.

This framework relies on measuring $\rho_{r_{AB}}$ and estimating $\rho^0_{r_{AB}}$. With protein structures, the best you can do for $\rho_{r_{AB}}$ is collect statistics from the protein data bank and pretend this is a statistical mechanical ensemble. For $\rho^0_{r_{AB}}$, the frequency you would expect by chance, one can use an analogy with chemistry, treat A and B as species in solution and consider the concentrations [A] and [B]. For proteins, you could treat the amino acid composition as if it was a mole fraction.

In practice, it may not be valid to treat proteins as if they were disconnected solutions of amino acids.(38) There might be artefacts due to packing effects and problems with the fictitious statistical mechanics.(39,40) It is hard to see what kind of ensemble a survey of the PDB really is, but it has been argued that the resulting numbers are Helmholtz free energies.(41,42) Pragmatically, it may not matter much how close these statistical score functions are to free energies. They definitely do reflect statistical tendencies within proteins and this may be all one needs for a threading application.(43) Despite the debate over details, the approach is clear. One takes a large set of proteins, collects statistics and converts them to a score function. One then expects this function to work well for proteins not included in its parameterisation.

If one believes the statistical mechanical basis for the statistics-based score functions, then one is dealing with a real energy which is properly calibrated against the rest of the world. There is, however, a quite different school of thought. If one is dealing with protein threading, or perhaps structure prediction in general, than maybe one need not be too concerned with real energies or reproducing the physics of protein folding. It is not important that a score function represent every false minimum or kinetic trap which a protein visits when folding. Instead, one wants a function which can distinguish between a correct and incorrect structure. This function will usually have some adjustable parameters and perhaps these can be optimised for protein fold recognition. (44-49) The result may be a function which does not look like a conventional model for energy, but formally is still a force field.

While this approach sounds attractive, it is not so simple to put in place. Firstly, one must select the underlying basis functions. In the literature, these have ranged from quasi-Lennard-Jones terms(44,48) to various kinds of sigmoidal function.(47,50) Next, there is a problem with the question as posed. We want to distinguish between correct and incorrect structures. We can say that the correct structure is whatever is given in the protein data bank, but unfortunately, there is almost an infinity of incorrect structures for a sequence and one would like the score function to penalise all of them. One way to encode this idea is to adopt a statistical approach and try to consider the distribution of incorrect structures.(50-53) Imagine you have some score function which produces an energy, E_X, for your sequence on a template structure X. If you generate a large number of incorrect or alternate structures, you can calculate their energies (E_{alt}). One convenient way is to take a sequence and put its residues onto every template you can find that is larger than the sequence. This guarantees that the alternative structures are protein-like. Next, you could plot out a histogram of the energies of the alternate structures as shown in Fig. 4(A). Empirically, the distribution of alternate energies (E_{alt}) looks like a Gaussian curve (50), and it can even be theoretically justified.(53). As well as the distribution of E_{alt}, Fig. 4(A) shows E_{nat}, the energy of the native structure. What we would like is to adjust the force field parameters so that E_{nat} is well separated from the mean energy of all the wrong structures, $\left\langle E_{alt} \right\rangle$. In other words, one wants to make $E_{nat} - \left\langle E_{alt} \right\rangle$ as large as possible, as shown in Fig. 4(B). At the same time, one does not want to simply scale the figure. Instead, one should keep the standard deviation of the distribution, $\sigma_{E_{alt}}$ as small as possible. This idea is captured by the standard statistical term, the z-score, given by

$$z - \text{score} = \frac{E_{nat} - \left\langle E_{alt} \right\rangle}{\sigma_{E_{alt}}}$$

So, with this philosophy, the aim is the find the score function which gives the greatest *z*-score. At the same time, the score function should not only work for one protein, it should work (ideally) for every protein it will ever be faced with. Then, the approach usually taken is to select a set of proteins for parameterisation and to adjust the force field to give the best *z*-score over the whole set. Numerically, one has to take the expression for the *z*-score, expand it in terms of the underlying energy expressions and parameters and use a numerical optimisation method to adjust the parameters. Hopefully, the score function will then work well, even with proteins it has never faced before.

4. Sequence to structure alignment

Given some kind of score function, there are two areas where it will be applied. Firstly, one needs the best arrangement of sequence residues on the template (sequence to structure alignment). Secondly, one needs a score function to rank the final structures, discussed in section 5. As discussed below, this may really lead to two different score functions.

Finding the best alignment of a sequence to a template structure is vital, but perhaps still a problem. In 1995, it was noted that sequence to structure alignments were typically error-prone.(54) More recently, the problem has been re-examined along with the consequences for fold recognition.(55) One can see the severity of the problem even with tiny errors. The average distance between C^{α} atoms is 3.8 Å, so a single residue mis-alignment would be enough to render a model useless in an application like drug design, even if the template molecule is close to correct for the unknown structure. Next, a larger misalignment, putting a gap of several residues at the wrong location, could easily send β-strand residues to a piece of α-helix or random coil. In the context of fold recognition, the problem is worse. Looking at Fig. 3, one can see that if the alignments are wrong, the models and calculated scores are wrong and it makes no sense to rank them.

There are two very clear reasons why sequence to structure alignment problems are difficult. The first is that the simple score functions commonly used are far from perfect. It is not practical to use the best atomistic force fields in the literature and the simpler, more coarse-grained ones cannot work as well. Next, the problem is formally difficult and in the most general case is NP-complete.(56) This can be explained by comparison with sequence to sequence comparison. Fig. 5A shows an alignment of "ACDEF" to some template which has both a sequence, "QRSTVW" and the structure shown. With the two gaps present, only three residues are aligned. If we consider a classic dynamic programming calculation for the alignment(57,58), we have to construct a scoring matrix as shown in Fig. 5B. The elements of the matrix reflect the similarity of amino acids. For example, the element indexed by "DR" comes from looking up the similarity of aspartate and arginine in a literature substitution matrix. The path marked on the score matrix corresponds to the alignment in part (A).

In contrast, consider the situation in Fig. 5(C). The same sequence is to be aligned, but now to a structure. One wants to construct a similar score matrix as shown on the right, but it is not possible. Looking again at the cell indexed by D2, one wants some kind of compatibility score. This implies the interactions shown on the left. While we can place residue D at position 2, the interactions with the other sites cannot be calculated since the other residues have not been aligned. For example, if one wants to calculate the interaction between sites 2 and 4, we may say that D is at place 2, but one does not know who it is interacting with at site 4. Clearly, sequence to structure alignments are routinely calculated so the problem is not impossible. It merely requires heuristics and approximations blending optimism, brute force and cunning.

One approach is to give up on dynamic programming completely. One has a score function and a discrete space, so the score of a trial alignment can always be calculated. In that case, the problem seems well suited to Monte Carlo / simulated annealing.(59) This, however, does not alleviate the problem of the huge search space. Allowing gaps and insertions at any position and of any length leads to a combinatorial explosion of possibilities. The calculation can be made tractable by restricting the search space and forbidding gaps except in recognised loops in template structures.(27,60)

In contrast, a dynamic programming approach has the advantage that it is deterministic and there are at least three approximations which squeeze a pair-wise, through space calculation into the framework of Fig. 5. Firstly, one could use the sequence of the template structure to start a process. In Fig. 5(C), the template has been drawn without its original sequence. One can, instead, leave the template residues in place. Then, to fill out an element in the score matrix such as the D2 position, one knows the interaction partners. For example, the first interaction could be calculated as D at position 2 with an S at position 3 where the S comes from the template sequence. After calculating a first alignment, the residue identities could be taken from the correct sequence and another alignment calculated. This method, usually known as the frozen approximation can be iterated a fixed number of times or until convergence.(36,61,62)

Another method for approximating the missing information was introduced by Jones *et al* in 1992 who used a second level of dynamic programming.(37) To continue to concentrate on one matrix element, one could conceptually place residue D at position 2 and then arrange the rest of the sequence to interact as favourably as possible. To score D at position 3, one would again recalculate the best arrangement of its sequence neighbours. This is still only an approximation to the correct answer and has been described as finding the best arrangement for every residue that it could have.(63)

Thirdly, one could modify the score function itself so as to make it suitable for a pairwise calculation. This can be done by using a score function which only uses the identity of one member of each interaction pair like DX, EX, FX, ... Furthermore, the score function can actually be optimised to work in this mean-field manner.(64)

Given the approximations necessary to treat the alignment problem by dynamic programming, several groups have developed branch and bound methods. Working in the phenomenally complex space of possible alignments, they try to successively rule out regions which cannot contribute to the answer.(65-68) The only disadvantage is that they tend to be slow and difficult to implement. A most recent version appears to be both swift and remarkably effective.(69) If more people were capable of programming these approaches, they might displace the ugly approximations in common use.

5. Fold recognition

If the steps described so far have been successful, one has a library of protein templates which is comprehensive and representative. There is a score function and a fast method for producing the best possible sequence to structure alignments and thus the best models possible. Unfortunately, the problem is still not solved. Imagine one has a library of 1 000 structures and only one of the templates is close to the correct answer. It is an act of faith to assume that the most correct model is the one that scored best during the alignment step.

One can introduce the problem of fold recognition by comparing it with a sequence database search. In that case, one assumes that the more similar a sequence, the more similar residues will have been a aligned and the higher the score will be. In protein threading, one use similar reasoning and says that only a similar template will provide a framework which lets the sequence residues interact favourably, so templates will score better, the closer they are to the correct answer. Unfortunately, it has been pointed out that the argument is not strictly valid(70). A sequence to structure alignment allows residues to move along non-physical degrees of freedom. In other words, a sequence to structure alignment may produce a favourable arrangement of residues in space, but it may not be one that occurs in nature. In practical terms, a good sequence to structure alignment method may arrange a sequence so that secondary structure is formed and hydrophobic residues will be close to each other, but on a completely wrong template.

Perhaps one should not even expect that one score function should be best for both arranging residues on a template and then ranking the models.(64,71) When calculating alignments, the score function is being asked which parts of a sequence are more suitable than other portions of the sequence for certain parts of a structure. When ranking models, one is asking a score function to rank the same sequence in different conformations. The situation, however, is even more interesting from a statistical point of view.

Imagine a score function which is useful for both sequence to structure alignments and recognising correct models and has no systematic bias or error. The only failing it has is a susceptibility to some quasi-random noise. In this case you might take the scores of your models, plot out the distribution and count the number of standard deviations (σ) that separate your best scores from the mean of the distribution. If your best score is 10 σ from the mean, it is almost certainly not a chance occurrence. If it is one σ from the mean, there is a significant chance that it is simply coincidental. Unfortunately, this approach, which was popular some years ago, is woefully inaccurate. The scores, especially when gaps and insertions are allowed, will be far from Gaussian distributed.

In pure sequence comparison, a mixture of theory and empiricism has been applied to assessing the significance of scores by estimating p-values (probability of a score occurring by chance) and e-values (expected number of times the score will be seen given the size of the database). For ungapped sequence comparisons, one can assume an extreme-value or Gumbel distribution(72-75). For gapped alignments, this is a useful approximation, but may not be absolutely correct.(76). For sequence to structure alignments, the problem is worse. As you add residues to a sequence, the score does not grow linearly. Instead, each residue you add may interact with its N-1 neighbours, so one might expect scores to grow with N^2. Unfortunately, the use of interaction cutoffs means that a conventional pair-wise interaction score is expected to grow as N^k where k is between 1 and 2, but varies depending on protein size. This means that the analytical formulae or regression approaches used in sequence comparison will not work with sequence to structure alignments.

Sommer *et al* actually had some success treating sequence to structure scores as if they followed a known distribution (77), but there is a different philosophy available. If one does not know what are the most important features of a reliable model are, one could instead take the likely descriptors and use a machine learning method to see what is useful. These would include the length of the sequence and template, the length of the alignment and various score function components.(78-81) As is common in neural networks, the approach is often effective, but not transparent. It is interesting that this could be interpreted as an example of using one function for sequence to structure alignments and a different one for ranking the resulting models.

With all these caveats, it is interesting to note that automatic prediction servers do give estimates of confidence in predictions. One should bear in mind that these are approximations and probably not as accurate as the statistical estimates for pure sequence comparison produced by programs such as BLAST, PSI-BLAST or FASTA(24-26).

6. Threading implementations and the broader context

To devotees, pure threading has an intellectual appeal. By using structural information, one should be able to detect similarities which are too weak to find by sequence based methods. With structural information, one would hope to find similarities even when there is no obvious evolutionary connection between a target sequence and close template. In practice, none of this may be true. Only a very brave spectator would name the best method for alignment and fold recognition, but it would be hard to argue that pure sequence-based methods are not amongst the very best. It is true that a simple sequence comparison method does not work well with weak homology, but current methods are much more advanced. Both PSI-BLAST(26) and the hidden Markov model methods(82-84) use families of related sequences, take advantage of the site-specific information found in a sequence alignment and the fact that although proteins A and B are not obviously similar, they are both reliably related to some protein C.

This probably does not spell the end of threading approaches. Instead, most threading approaches now incorporate information beyond pure through-space scoring information. For example, consider again Fig. 5 which shows a score matrix for some sequence against a template of known sequence and structure. Fig 5(B) and (C) show different score matrices from the sequence-sequence and sequence-structure terms. If they offer independent information, there is no reason they cannot be combined. This implies some weighting of the different terms either by trial and error (85) or even by applying a numerical optimisation method.(86). Rather than simply add in a sequence term, one can take advantage of the profiles of sequences related to the sequence of interest, the library template or both.(81,85,87) Obviously, combining sequence to sequence and sequence to structure terms is only useful if they contain independent information, but all the proponents would assert that they do.

This idea can be extended to other kinds of information. The secondary structure of a template is easily calculated. If one could reliably predict the secondary structure of a sequence, one could match it to the template. Even without perfect secondary structure predictions, they certainly provide more signal than noise and are routinely added to threading calculations.(78,81,87-95)

7. Context, application and obsolescence

Given the selection of methods in the literature, threading means different things to different groups. Whatever the definition, is it ever useful and are there places where it should be avoided ? For the sake of argument, one can call threading some method which implements a through-space scoring function, combined with some of the terms from section 6 and performs sequence to structure alignments.

Firstly, one can say that threading must produce better alignments than methods using only sequence information. This is true, because structure only adds information. If it is not true in practice, it means that implementations are not optimal or one is not making a fair comparison. Pure sequence based methods with profiles are extremely sensitive in finding remote homologues. Pure outdated threading methods are not as sensitive. Newer threading methods use sequence profiles and have absorbed many of the methods of sequence analysis. They certainly should not do worse than any other method.

Next, can one define areas where threading should be the technique of choice ? The ideal problem for a threading-partisan is

- a sequence of unknown structure
- the sequence should have no detectable homology to anything of known structure
- there should be a known structure which is very similar to the unknown
- there should be no functional clues as to the structural class, otherwise a biochemist may recognise the similarity

This situation can occur and it is not always recognised. A more likely scenario is that the borders are blurred and the thresholds uncertain. There may be some functional information about a sequence, but a chemist would like confirmation of beliefs or reassurance from a calculation. Sequence searches may have suggested plausible homologues of known structure, but with too little statistical confidence to be reliable.

One may not be obliged to follow a threading procedure as a fixed recipe. If sequence searches have suggested structural templates, but of very low sequence identity, then a sequence to structure alignment may be a useful step in building a model. This would not count as a threading calculation, but would use methods developed under the methodological umbrella of threading.

Changing viewpoint, can one identify times when threading should be avoided ? If a sequence has very high homology to something of known structure, then threading should not do any harm, but may be a waste of time. Occasionally, however, the additional information from through-space score functions will not be helpful. If a protein has unusual structural properties, they may not be well modelled in the simple scoring functions commonly used. For example, calculations on a protein which seems to have no structure in the absence of a cofactor or prosthetic group may produce a disaster. Membrane bound proteins are also a special problem, since most low resolution force fields implicitly assume water is the solvent. Even simple factors such as size may be important. Small proteins may be disulfide rich or problematic simply because of their large surface to volume ratio.

Maybe the question of when to thread is not really a problem. Threading calculations should be cheap and one does not have to use or believe the results. They are also not difficult to run. One can either find a relevant code and run it locally or use one of the web sites which provide an interface to several methods and even an assessment of the different implementations.(96-98)

If one is worried about the reliability of answers, one can also look for an area where some errors are tolerated. If you are interested in genome scale applications, it is a natural consequence that you will accept a finite error rate, perhaps using threading calculations as just part of a larger computational pipeline for screening sequences.(81,99,100) Furthermore, there will even be applications where the exact structure is not important. If one wants to pick targets for structural genomics, one may try to find those sequences whose structure is most difficult to predict. Again, protein threading may be one of the tools used.(101)

Since threading has already changed since the early implementations, it is also clear that the methods will continue to evolve. Some techniques combine elements of threading with methods for de novo structure prediction(102-105). This holds the promise of being able to predict structures unlike any previously solved. Threading may also be applied in new contexts such as macromolecular interactions and multimolecular assemblies.(100,106)

In the absence of any intellectual progress, the simple accumulation of experimental data makes prediction methods work better. The raw bulk of sequence data means that sequence profiles built now are almost always better than a year ago. This alone helps threading calculations which use sequence profiles. At the same time, the growth of the protein data bank means that there is an ever increasing chance of a structural homologue existing for the sequence of interest.

Probably the most frightening prospect for an advocate of pure threading has been the occasional success of some fragment assembly methods and their remarkable predictions, even for previously unseen folds.(107-110) If methods for *ab initio* or *de novo* structure prediction become reliable, protein threading will be obsolesced without ever really having had a phase of glory.

References

1. Levitt, M. (1975) Computer simulation of protein folding. *Nature* **253**, 694-698.
2. Levitt, M. (1976) A simplified representation of protein conformations for rapid simulation of protein folding. *J. Mol. Biol.* **104**, 59-107.
3. Crippen, G. M. & Maiorov, V. N. (1995) How many protein-folding motifs are there. *J. Mol. Biol.* **252**, 144-151.
4. Leonov, H., Mitchell, J. S. B. & Arkin, I. T. (2003) Monte Carlo estimation of the number of possible protein folds: Effects of sampling bias and folds distributions. *Proteins* **51**, 352-359.
5. Wolf, Y. I., Grishin, N. V. & Koonin, E. V. (2000) Estimating the number of protein folds and families from complete genome data. *J. Mol. Biol.* **299**, 897-905.
6. Govindarajan, S., Recabarren, R. & Goldstein, R. K. (1999) Estimating the total number of protein folds. *Proteins* **35**, 408-414.
7. Zhang, C. O. & DeLisi, C. (1998) Estimating the number of protein folds. *J. Mol. Biol.* **284**, 1301-1305.
8. Wang, Z. X. (1998) A re-estimation for the total numbers of protein folds and superfamilies. *Protein Eng.* **11**, 621-626.
9. Zhang, C. T. (1997) Relations of the numbers of protein sequences, families and folds. *Protein Eng.* **10**, 757-761.
10. Wang, Z. X. (1996) How many fold types of protein are there in nature? *Proteins* **26**, 186-191.
11. Orengo, C. A., Jones, D. T. & Thornton, J. M. (1994) Protein superfamilies and domain superfolds. *Nature* **372**, 631-634.
12. Chothia, C. (1992) Proteins - 1000 families for the molecular biologist. *Nature* **357**, 543-544.

13. England, J. L., Shakhnovich, B. E. & Shakhnovich, E. I. (2003) Natural selection of more designable folds: A mechanism for thermophilic adaptation. *Proc. Natl. Acad. Sci. U.S.A.* **100**, 8727-8731.

14. Li, H., Tang, C. & Wingreen, N. S. (2002) Designability of protein structures: A lattice-model study using the miyazawa-jernigan matrix. *Proteins* **49**, 403-412.

15. Miller, J., Zeng, C., Wingreen, N. S. & Tang, C. (2002) Emergence of highly designable protein-backbone conformations in an off-lattice model. *Proteins* **47**, 506-512.

16. Helling, R., Li, H., Melin, R., Miller, J., Wingreen, N., Zeng, C. & Tang, C. (2001) The designability of protein structures. *Journal of Molecular Graphics & Modelling* **19**, 157-167.

17. Shahrezaei, V. & Ejtehadi, M. R. (2000) Geometry selects highly designable structures. *J. Chem. Phys.* **113**, 6437-6442.

18. Bornberg-Bauer, E. (1997) How are model protein structures distributed in sequence space? *Biophysical Journal* **73**, 2393-2403.

19. Govindarajan, S. & Goldstein, R. A. (1996) Why are some protein structures so common? *Proc. Natl. Acad. Sci. U.S.A.* **93**, 3341-3345.

20. Orengo, C. (1994) Classification of protein folds. *Curr. Opin. Struct. Biol.* **4**, 429-440.

21. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000) The protein data bank. *Nucleic Acids Res* **28**, 235-42.

22. Brenner, S. E., Chothia, C. & Hubbard, T. J. P. (1998) Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc. Natl. Acad. Sci. U.S.A.* **95**, 6073-6078.

23. Rost, B. (1999) Twilight zone of protein sequence alignments. *Protein Eng.* **12**, 85-94.

24. Pearson, W. & Lipman, D. (1988) Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. U.S.A.* **85**, 2444-2448.

25. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990) Basic local alignment search tool. *J. Mol. Biol.* **215**, 403-410.

26. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997) Gapped blast and psi-blast: A new generation of protein database search programs. *Nucleic Acids Res* **25**, 3389-3402.

27. Madej, T., Gibrat, J. F. & Bryant, S. H. (1995) Threading a database of protein cores. *Proteins* **23**, 356-369.

28. Huber, T. & Torda, A. E. (2002) Protein structure prediction by threading: Force field philosophy, approaches to alignment. In *Protein structure prediction: A bioinformatic approach* (Tsigelny, I. F., ed.). International University Line, La Jolla, pp. 263-298.

29. Cornell, W. D., Cieplak, P., Bayly, C. I., Gould, I. R., Merz, K. M., Ferguson, D. M., Spellmeyer, D. C., Fox, T., Caldwell, J. W. & Kollman, P. A. (1995) A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J. Am. Chem. Soc.* **117**, 5179-5197.

30. van Gunsteren, W. F., Billeter, S. R., Eising, A. A., Huenenberger, P. H., Krueger, P., Mark, A., Scott, W. R. P. & Tironi, I. G. (1996) Biomolecular simulation: The gromos96 manual and user guide, vdf Hochschulverlag AG an der ETH Zurich and BIOMOS b.v., Zurich and Groningen.

31. MacKerell, A. D., Bashford, D., Bellott, M., Dunbrack, R. L., Evanseck, J. D., Field, M. J., Fischer, S., Gao, J., Guo, H., Ha, S., Joseph-McCarthy, D., Kuchnir, L., Kuczera, K., Lau, F. T. K., Mattos, C., Michnick, S., Ngo, T., Nguyen, D. T., Prodhom, B., Reiher, W. E., Roux, B., Schlenkrich, M., Smith, J. C., Stote, R., Straub, J., Watanabe, M., Wiorkiewicz-Kuczera, J., Yin, D. & Karplus, M. (1998) All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J. Phys. Chem. B* **102**, 3586-3616.

32. Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S. & Karplus, M. (1983) Charmm - a program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.* **4**, 187-217.

33. Chandler, D. (1987) Introduction to modern statistical mechanics, Oxford University Press, New York.

34. Miyazawa, S. & Jernigan, R. L. (1985) Estimation of effective interresidue contact energies from protein crystal structures: Quasi-chemical approximation. *Macromolecules* **18**, 534-552.

35. Tanaka, S. & Scheraga, H. A. (1976) Statistical mechanical treatment of protein conformation. 1. Conformational properties of amino-acids in proteins. *Macromolecules* **9**, 142-159.

36. Sippl, M. J. (1993) Boltzmann's principle, knowledge-based mean fields and protein folding. An approach to the computational determination of protein structures. *J Comput Aided Mol Des* **7**, 473-501.

37. Jones, D. T., Taylor, W. R. & Thornton, J. M. (1992) A new approach to protein fold recognition. *Nature* **358**, 86-9.

38. Skolnick, J., Jaroszewski, L., Kolinski, A. & Godzik, A. (1997) Derivation and testing of pair potentials for protein folding. When is the quasichemical approximation correct? *Protein Sci.* **6**, 676-688.

39. Ben-Naim, A. (1997) Statistical potentials extracted from protein structures: Are these meaningful potentials ? *J. Chem. Phys.* **107**, 3698-3706.

40. Thomas, P. D. & Dill, K. (1996) Statistical potentials extracted from protein structures: How accurate are they ? *J. Mol. Biol.* **257**, 457-469.

41. Sippl, M. J. (1996) Helmholtz free energy of peptide hydrogen bonds in proteins. *J. Mol. Biol.* **260**, 644-8.

42. Sippl, M. J., Ortner, M., Jaritz, M., Lackner, P. & Flockner, H. (1996) Helmholtz free energies of atom pair interactions in proteins. *Fold. Des.* **1**, 289-98.

43. Shortle, D. (2003) Propensities, probabilities, and the boltzmann hypothesis. *Protein Sci.* **12**, 1298-1302.

44. Crippen, G. M. & Snow, M. E. (1990) A 1.8 angstrom resolution potential function for protein folding. *Biopolymers* **29**, 1479-1489.

45. Crippen, G. M. (1996) Easily searched protein folding potentials. *J. Mol. Biol.* **260**, 467-75.

46. Goldstein, R. A., Luthey-Schulten, Z. A. & Wolynes, P. G. (1992) Protein tertiary structure recognition using optimized hamiltonians with local interactions. *Proc Natl Acad Sci USA* **89**, 9029-9033.

47. Maiorov, V. N. & Crippen, G. M. (1992) Contact potential that recognizes the correct folding of globular-proteins. *J. Mol. Biol.* **227**, 876-888.

48. Seetharamulu, P. & Crippen, G. M. (1991) A potential function for protein folding. *J. Math. Chem.* **6**, 91-110.

49. Ulrich, P., Scott, W., van Gunsteren, W. F. & Torda, A. E. (1997) Protein structure prediction force fields - parametrization with quasi-newtonian dynamics. *Proteins* **27**, 367-384.

50. Huber, T. & Torda, A. E. (1998) Protein fold recognition without boltzmann statistics or explicit physical basis. *Protein Sci.* **7**, 142-149.

51. Hao, M. H. & Scheraga, H. A. (1996) How optimization of potential functions affects protein folding. *Proc. Natl. Acad. Sci. U.S.A.* **93**, 4984-4989.

52. Mirny, L. A. & Shakhnovich, E. I. (1996) How to derive a protein folding potential - a new approach to an old problem. *J. Mol. Biol.* **264**, 1164-1179.

53. Koretke, K. K., Luthey-Schulten, Z. & Wolynes, P. G. (1996) Self-consistently optimized statistical mechanical energy functions for sequence structure alignment. *Protein Sci.* **5**, 1043-1059.

54. Lemer, C. M., Rooman, M. J. & Wodak, S. J. (1995) Protein structure prediction by threading methods: Evaluation of current techniques. *Proteins* **23**, 337-355.

55. Chang, J., Carrillo, M. W., Waugh, A., Wei, L. P. & Altman, R. B. (2002) Scoring functions sensitive to alignment error have a more difficult search: A paradox for threading. In *Structures and mechanisms: From ashes to enzymes*, Vol. 827, pp. 309-320.

56. Lathrop, R. H. (1994) The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng.* **7**, 1059-1068.

57. Smith, T. F. & Waterman, M. S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.* **147**, 195-197.

58. Needleman, S. B. & Wunsch, C. D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48**, 443-453.

59. Kirkpatrick, S., Gelatt Jr., C. D. & Vecchi, M. P. (1983) Optimization by simulated annealing. *Science* **220**, 671-680.

60. Bryant, S. H. & Lawrence, C. E. (1993) An empirical energy function for threading protein-sequence through the folding motif. *Proteins* **16**, 92-112.

61. Wilmanns, M. & Eisenberg, D. (1995) Inverse protein folding by the residue pair preference profile method: Estimating the correctness of alignments of structurally compatible sequences. *Protein Eng* **8**, 627-639.

62. Godzik, A., Kolinski, A. & Skolnick, J. (1992) Topology fingerprint approach to the inverse protein folding problem. *J. Mol. Biol.* **227**, 227-238.

63. Taylor, W. R. (1997) Multiple sequence threading: An analysis of alignment quality and stability. *J. Mol. Biol.* **269**, 902-943.

64. Huber, T. & Torda, A. E. (1999) Protein sequence threading, the alignment problem and a two step strategy. *J. Comput. Chem.* **20**, 1455-1467.

65. Xu, Y. & Xu, D. (2000) Protein threading using prospect: Design and evaluation. *Proteins* **40**, 343-354.

66. Lathrop, R. H. (1999) An anytime local-to-global optimization algorithm for protein threading in o(m(^{2})n(^{2})) space. *J. Comput. Biol.* **6**, 405-418.

67. Xu, Y. & Uberbacher, E. C. (1996) A polynomial-time algorithm for a class of protein threading problems. *Comput. Appl. Biosci.* **12**, 511-517.

68. Lathrop, R. H. & Smith, T. F. (1996) Global optimum protein threading with gapped alignment and empirical pair score functions. *J. Mol. Biol.* **255**, 641-665.

69. Xu, J. & Li, M. (2003) Assessment of raptor's linear programming approach in cafasp3. *Proteins* **53**, 579-584.

70. Crippen, G. M. (1996) Failures of inverse folding and threading with gapped alignment. *Proteins* **26**, 167-171.

71. Park, B. H., Huang, E. S. & Levitt, M. (1997) Factors affecting the ability of energy functions to discriminate correct from incorrect folds. *J. Mol. Biol.* **266**, 831-846.

72. Altschul, S. F., Boguski, M. S., Gish, W. & Wootton, J. C. (1994) Issues in searching molecular sequence databases. *Nature Genetics* **6**, 119-129.

73. Altschul, S. F. & Gish, W. (1996) Local alignment statistics. In *Methods enzymol.*, Vol. 266, pp. 460-480.

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

75. Pearson, W. R. (1998) Empirical statistical estimates for sequence similarity searches. *J. Mol. Biol.* **276**, 71-84.

76. Mott, R. (2000) Accurate formula for p-values of gapped local sequence and profile alignments. *J. Mol. Biol.* **300**, 649-659.

77. Sommer, I., Zien, A., von Ohsen, N., Zimmer, R. & Lengauer, T. (2002) Confidence measures for protein fold recognition. *Bioinformatics* **18**, 802-812.

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

79. Juan, D., Grana, O., Pazos, F., Fariselli, P., Casadio, R. & Valencia, A. (2003) A neural network approach to evaluate fold recognition results. *Proteins* **50**, 600-608.

80. Xu, Y., Xu, D. & Olman, V. (2002) A practical method for interpretation of threading scores: An application of neural network. *Stat. Sin.* **12**, 159-177.

81. McGuffin, L. J. & Jones, D. T. (2003) Improvement of the genthreader method for genomic fold recognition. *Bioinformatics* **19**, 874-881.

82. Karplus, K., Sjolander, K., Barrett, C., Cline, M., Haussler, D., Hughey, R., Holm, L. & Sander, C. (1997) Predicting protein structure using hidden markov models. *Proteins*, 134-139.

83. Karplus, K., Barrett, C. & Hughey, R. (1998) Hidden markov models for detecting remote protein homologies. *Bioinformatics* **14**, 846-856.

84. Karplus, K., Barrett, C., Cline, M., Diekhans, M., Grate, L. & Hughey, R. (1999) Predicting protein structure using only sequence information. *Proteins*, 121-125.

85. Panchenko, A. R., Marchler-Bauer, A. & Bryant, S. H. (2000) Combination of threading potentials and sequence profiles improves fold recognition. *J. Mol. Biol.* **296**, 1319-1331.

86. Russell, A. & Torda, A. E. (2002) Protein sequence threading - averaging over structures. *Proteins* **47**, 496-505.

87. Kelley, L. A., MacCallum, R. M. & Sternberg, M. J. E. (2000) Enhanced genome annotation using structural profiles in the program 3D-PSSM. *J. Mol. Biol.* **299**, 499-520.

88. Fischer, D. & Eisenberg, D. (1996) Protein fold recognition using sequence-derived predictions. *Protein Sci.* **5**, 947-955.

89. Russell, R. B., Copley, R. R. & Barton, G. J. (1996) Protein fold recognition by mapping predicted secondary structures. *J. Mol. Biol.* **259**, 349-365.

90. Rost, B., Schneider, R. & Sander, C. (1997) Protein fold recognition by prediction-based threading. *J. Mol. Biol.* **270**, 471-480.

91. Di Francesco, V., Munson, P. J. & Garnier, J. (1999) Foresst: Fold recognition from secondary structure predictions of proteins. *Bioinformatics* **15**, 131-140.

92. Ayers, D. J., Gooley, P. R., Widmer-Cooper, A. & Torda, A. E. (1999) Enhanced protein fold recognition using secondary structure information from NMR. *Protein Sci.* **8**, 1127--1133.

93. Hargbo, J. & Elofsson, A. (1999) Hidden markov models that use predicted secondary structures for fold recognition. *Proteins* **36**, 68-76.

94. Ota, M., Kawabata, T., Kinjo, A. R. & Nishikawa, K. (1999) Cooperative approach for the protein fold recognition. *Proteins*, 126-132.

95. Koretke, K. K., Russell, R. B., Copley, R. R. & Lupas, A. N. (1999) Fold recognition using sequence and secondary structure information. *Proteins*, 141-148.

96. Rost, B. & Liu, J. F. (2003) The predictprotein server. *Nucleic Acids Res* **31**, 3300-3304.

97. Eyrich, V. A. & Rost, B. (2003) Meta-pp: Single interface to crucial prediction servers. *Nucleic Acids Res* **31**, 3308-3310.

98. Koh, I. Y. Y., Eyrich, V. A., Marti-Renom, M. A., Przybylski, D., Madhusudhan, M. S., Eswar, N., Grana, O., Pazos, F., Valencia, A., Sali, A. & Rost, B. (2003) Eva: Evaluation of protein structure prediction servers. *Nucleic Acids Res* **31**, 3311-3315.

99. Kim, D., Xu, D., Guo, J. T., Ellrott, K. & Xu, Y. (2003) Prospect II: Protein structure prediction program for genome-scale applications. *Protein Eng.* **16**, 641-650.

100. Lu, L., Lu, H. & Skolnick, J. (2002) Multiprospector: An algorithm for the prediction of protein- protein interactions by multimeric threading. *Proteins* **49**, 350-364.

101. McGuffin, L. J. & Jones, D. T. (2002) Targeting novel folds for structural genomics. *Proteins* **48**, 44-52.

102. Jones, D. T. (2001) Predicting novel protein folds by using fragfold. *Proteins*, 127-132.

103. Skolnick, J., Kolinski, A., Kihara, D., Betancourt, M., Rotkiewicz, P. & Boniecki, M. (2001) Ab initio protein structure prediction via a combination of threading, lattice folding, clustering, and structure refinement. *Proteins*, 149-156.

104. Zhang, Y., Kolinski, A. & Skolnick, J. (2003) Touchstone II: A new approach to ab initio protein structure prediction. *Biophysical Journal* **85**, 1145-1164.

105. Kihara, D., Lu, H., Kolinski, A. & Skolnick, J. (2001) Touchstone: An ab initio protein structure prediction method that uses threading-based tertiary restraints. *Proc. Natl. Acad. Sci. U.S.A.* **98**, 10125-10130.

106. Lu, L., Arakaki, A. K., Lu, H. & Skolnick, J. (2003) Multimeric threading-based prediction of protein-protein interactions on a genomic scale: Application to the saccharomyces cerevisiae proteome. *Genome Research* **13**, 1146-1154.

107. Simons, K. T., Strauss, C. & Baker, D. (2001) Prospects for ab initio protein structural genomics. *J. Mol. Biol.* **306**, 1191-1199.

108. Simons, K. T., Kooperberg, C., Huang, E. & Baker, D. (1997) Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J. Mol. Biol.* **268**, 209-225.

109. Chivian, D., Robertson, T., Bonneau, R. & Baker, D. (2003) Ab initio methods. In *Structural bioinformatics* (Bourne, P. E. & Weissig, H., eds.), Vol. 44. Wiley-Liss, Hoboken, N.J., pp. 547-548.

110. Bonneau, R. & Baker, D. (2001) Ab initio protein structure prediction: Progress and reports. *Annu Rev Biophys Biomol Struct* **30**, 173-189.

Figure Captions

Fig. 1. A sequence of unknown structure and a template library (collection of known structures)

Fig. 2. Threading and aligning a sequence through a template library.

Fig. 3. Set of candidate structures for sequence. Models correspond to alignments from Fig. 2

Fig. 4. Z-score optimisation for force field construction. E_{alt} is the energy of an alternative conformation; E_{nat} the energy of the native (correct) structure; $\langle E_{alt} \rangle$ the mean of the alternative conformation energies; N_{alt} the number of alternative conformations of a given energy. In (B), $\sigma_{E_{alt}}$ is the standard deviation of the E_{alt} distribution; E_{nat} - $\langle E_{alt} \rangle$ is the difference between the energy of the native conformation and the average of alternate conformation energies.

Fig. 5. Comparison of sequence to sequence and sequence to structure alignments.

Figure 1, Torda chapter, proteomics handbook

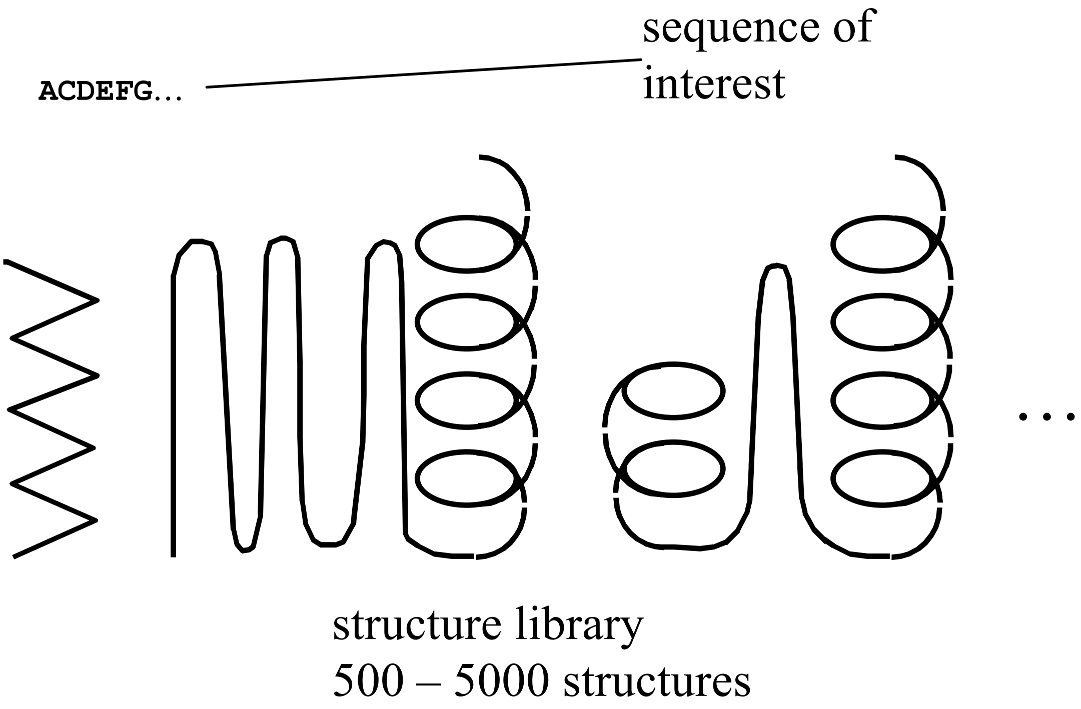


Figure 2, Torda chapter, Proteomics Handbook

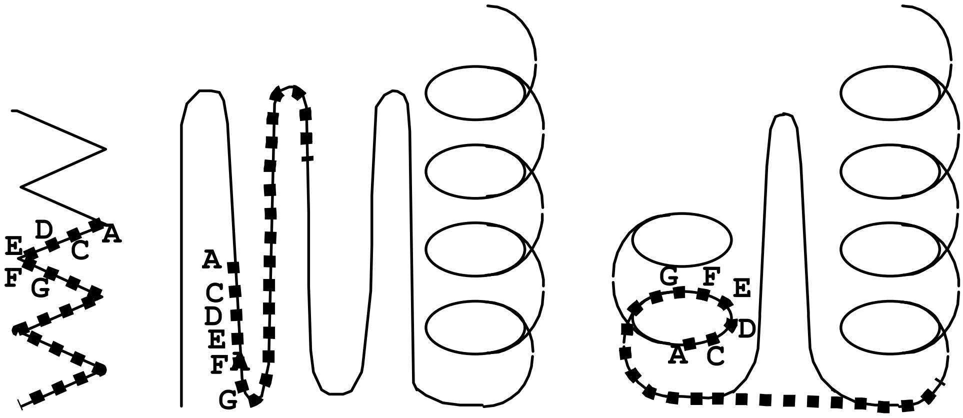


Figure 3, Torda chapter, Proteomics Handbook

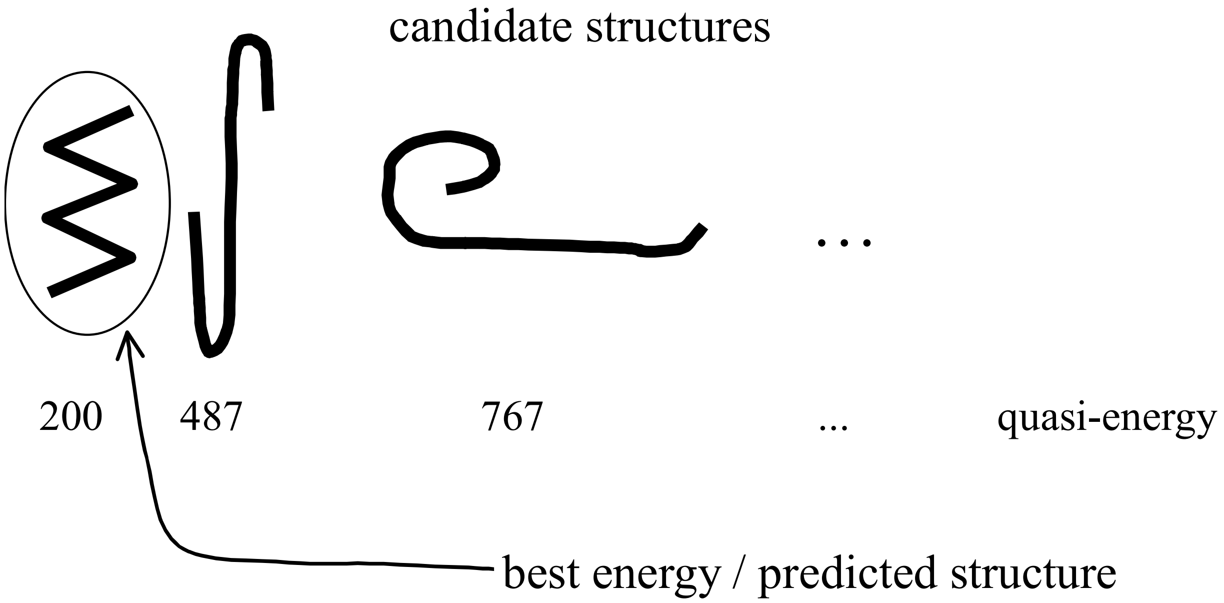


Figure 4 , Torda chapter, Proteomics Handbook

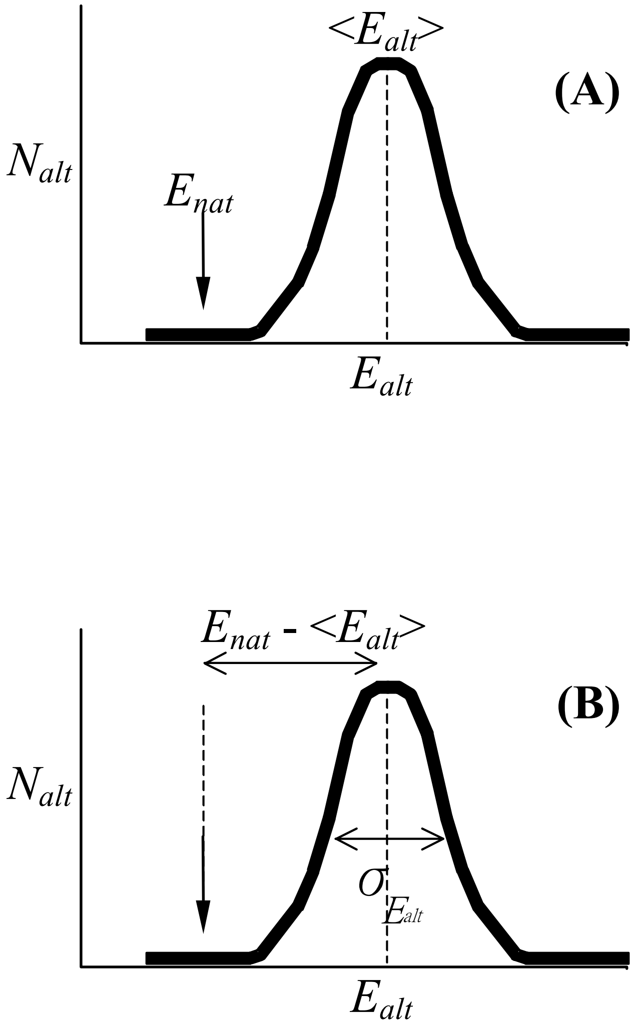
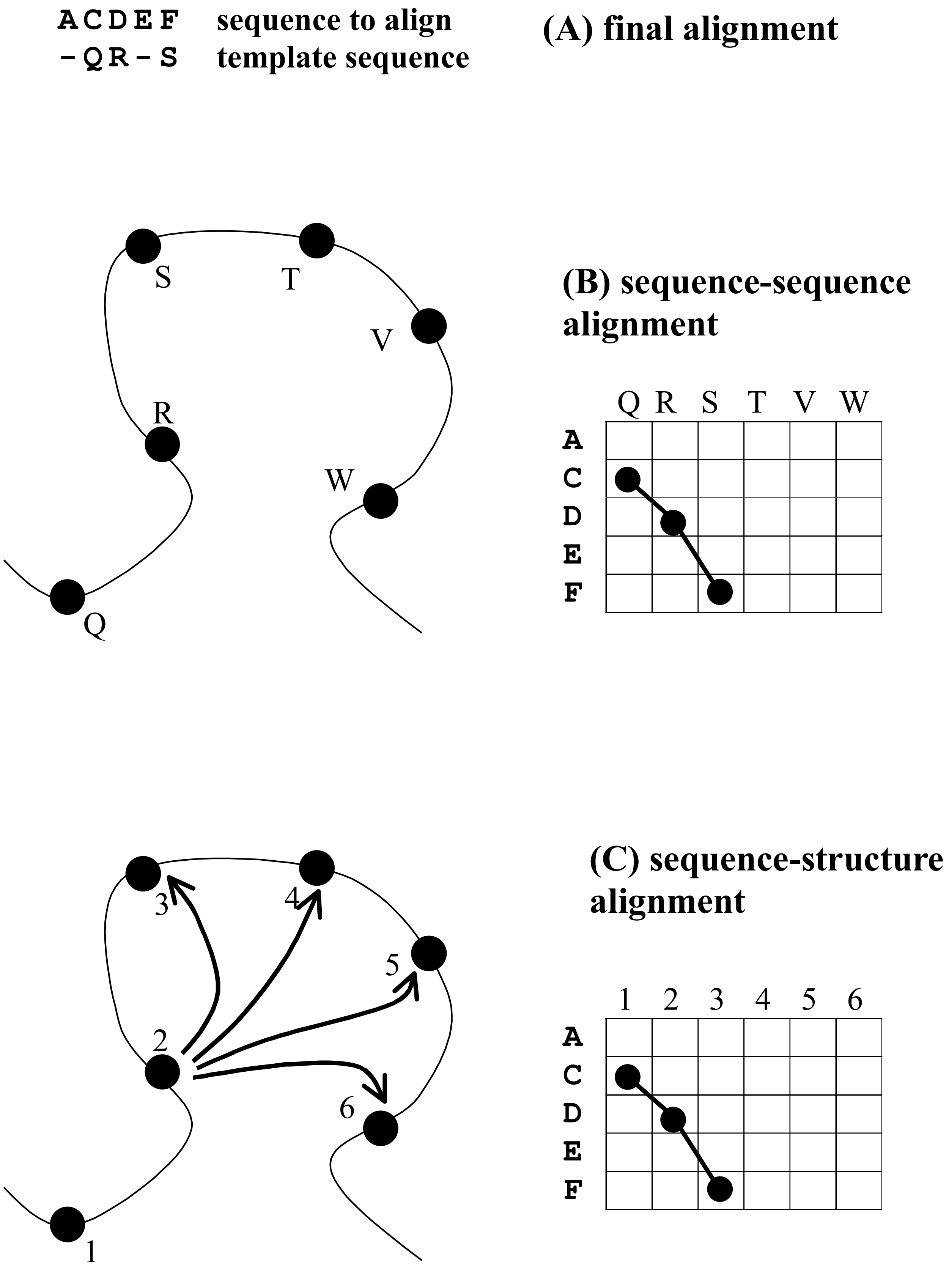


Figure 5 , Torda chapter, Proteomics Handbook



```
ACDEF   sequence to align
-QR-S   template sequence
```

(A) final alignment

(B) sequence-sequence alignment

(C) sequence-structure alignment