CSIRO Publishing

AUSTRALIAN JOURNAL OF CHEMISTRY

AN INTERNATIONAL JOURNAL FOR CHEMICAL SCIENCE

publishing research papers from all fields of chemical science, including synthesis, structure, new materials, macromolecules, supramolecular chemistry, biological chemistry, nanotechnology, surface chemistry, and analytical techniques.

Volume 54, 2001

All enquiries and manuscripts should be directed to:

Dr Alison Green
Australian Journal of Chemistry – an International Journal for Chemical Science
CSIRO PUBLISHING
PO Box 1139 (150 Oxford St)
Collingwood, Vic. 3066, Australia

Telephone: +61 3 9662 7630
Fax: +61 3 9662 7611
E-mail: publishing.ajc@csiro.au

Published by CSIRO PUBLISHING
for CSIRO and the Australian Academy of Science

www.publish.csiro.au/journals/ajc

Aust. J. Chem. **2001**, *54*, 367–373

Comparing Objects of Different Sizes: Treating Proteins as Strings

James B. Procter,[A] *Andrew J. Perry*[B] *and Andrew E. Torda*[A,C]

[A] Research School of Chemistry, Australian National University, Canberra, A.C.T. 0200, Australia.
[B] The Russell Grimwade School of Biochemistry and Molecular Biology,
The University of Melbourne, Melbourne, Vic. 3010, Australia.
[C] Author to whom corresponsence should be addressed (e-mail: andrew.torda@anu.edu.au).

For low-resolution force field calculations on proteins, one needs a method to automatically identify weak structural similarities in different molecules. A method is presented which treats sets of three-dimensional coordinates as graphs, and generates a similarity matrix based on local clique matching density that is searched by dynamic programming to identify similarities in the original structures. The method meets the minimal requirements that it works with protein molecules of different sizes and works with gaps and insertions. Examples are given where similarities between protein structures have been detected, despite no similarity being detectable by simple inspection.

Manuscript received: 12 October 2000.
Final version: 15 October 2001.

Introduction

Molecular modelling is a broad term. To a quantum chemist, it may mean debating fractions of an angstrom, but to a polymer physicist, it could mean guessing a system's typical radius of gyration, if only to the nearest hundred angstrom. At some intermediate level of detail, one may find low-resolution protein calculations. In these, the coordinates of some atoms may be known to within a few angstroms while others are known with extremely little certainty. Typically, such models are built using a mixture of coarse-grained molecular mechanics and deduction from evolutionary relations. To a molecular biologist, this kind of low-resolution structure may be quite useful. If it correctly predicts which parts of a molecule are solvent exposed or candidates for interacting with other molecules, it may be sufficient to guide site-directed mutagenesis work. In some cases, this rough model may be sufficient to identify chemically characteristic conformations—allowing the further deduction of biochemical properties, and even suggesting a lead for rational drug design. However, such feats typically require a combination of extensive experience, and a source of reliable structural data. Therefore, at the very least, one must be able to judge the quality of such models, before initiating a costly investigation based on potentially incorrect data.

For molecular mechanics calculations there are standard methods for dealing with low-resolution representations. In protein or polymer simulations one often works with 'united atoms' where a single interaction site represents several particles. This is perhaps because such a reduced representation is sufficient for the model of behaviour, but is most commonly due to the size of these systems. However, when relating these results to those in other areas, new problems can arise. Specifically in the case of proteins, one often works with the concept of a protein fold, which refers to some overall shape and arrangement of the backbone of the polymer chain. This may be meaningful to a molecular biologist, but seems too indistinct for numerical work. A rigorous definition is elusive because a group of molecules with a common fold that can be readily categorized by an expert usually contains a variety of different sized proteins. With such ambiguity, it is unsurprising that the various structure classification methods are not in accord.[1] It seems clear that this disagreement can only be resolved by a more complete understanding of protein similarity, but this is not simply to satisfy scientific pedantry. Determination of the type of structure (fold) a sequence is most likely to adopt in vivo is a common goal of computational chemistry,[2] and a key to the interpretation of the mass of sequence data currently being produced by genomic programs.

In order to work with protein folds and their prediction, one needs measures of similarity between proteins. These are used to build parameterization sets that can facilitate fold prediction, and to measure the similarity of predictions to known answers enabling the assessment of their reliability. Any method for determining this measure must be able to compare proteins of different sizes, and detect weak

 10.1071/CH01053 0004-9425/01/060367

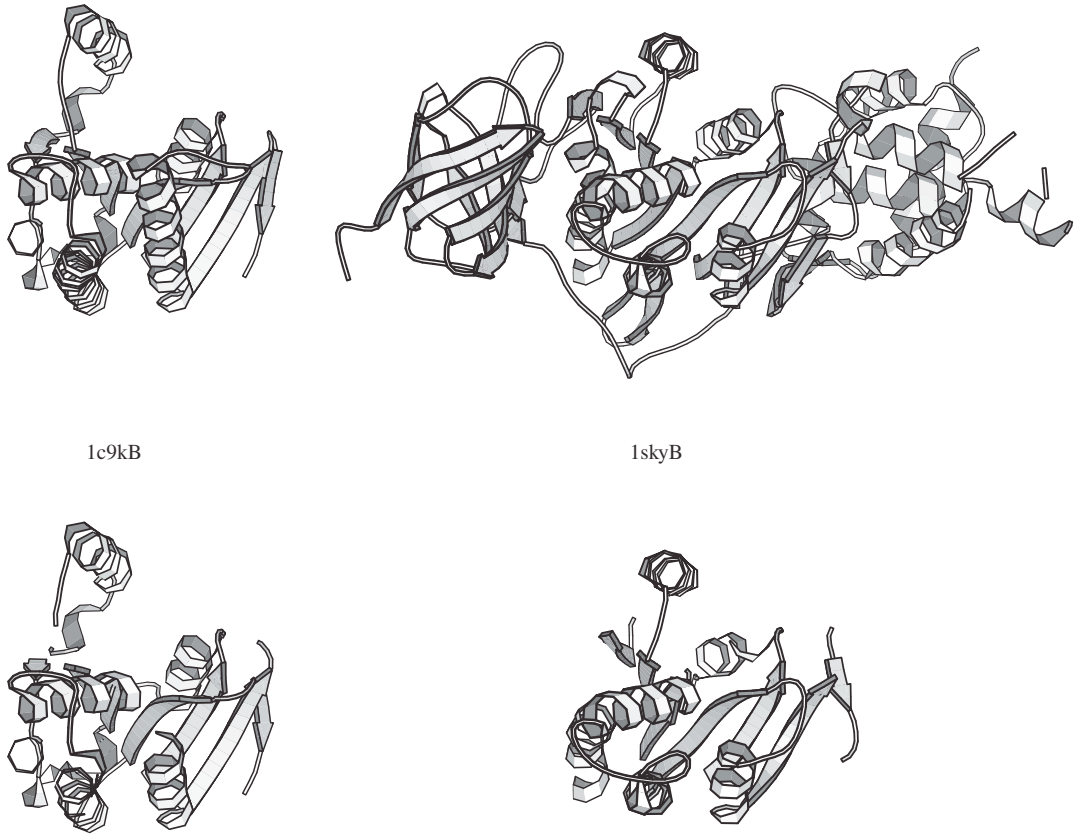

Fig. 1. A comparison of two proteins: the 'B' chains of 1c9k, a guanylyl transferase, and 1sky, an ATP synthase with three structural domains. The full structures are shown above the labels, and the common cores are shown below. Schematics produced with MOLSCRIPT.[3]

similarities when only parts of either molecule have any resemblance. Worse than that, the similar regions of proteins may not even be continuous in terms of the protein backbone. In this paper, we present a method that can deal with these problems and automatically locate similar regions in otherwise dissimilar structures.

To see the severity of the problem, consider the pair of structures shown in Fig. 1. As shown, the similarity between the two proteins is clear, but a human expert may have problems identifying it if the coordinates were not oriented and the similar regions highlighted. From the viewpoint of automatic methods, it is easy to see why the problem is so difficult. If one were to stretch out each polymer/protein chain as shown in Fig. 2, the similar regions could only be found by introducing gaps in the backbone chains. If one is willing to introduce gaps of any length at any position in either protein, the search space grows astronomically. This leads to another complication. If one inserts a huge number of gaps it becomes easier to find matching regions. At the same time, the intervening segments become very small and eventually structurally meaningless. For this reason, one has to introduce the idea of a gap penalty or cost associated with introducing a break into one of the chains, allowing the formulation of a numerical optimization problem.

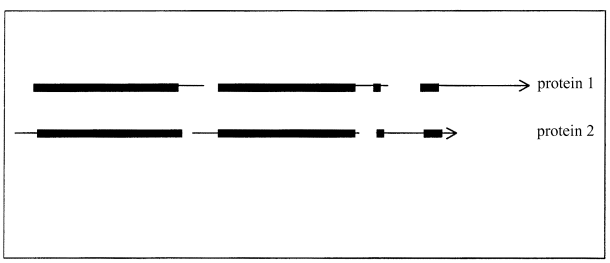

Fig. 2. Diagram of two protein chains after structural alignment. Each broken arrow represents a protein chain. The thick regions show where structural similarity is found.

Although the most general case of different object comparison may be intractable, there are many useful heuristics in common use. None may be guaranteed to give optimal answers or work in every case. All will have some arbitrary thresholds. First, one can take advantage of regularities in protein structure to simplify the problem. Practically all approaches start from the fact that proteins are unbranched polymers formed from a set of standard monomers, so the chain may be adequately characterized by the set of C^{α} atoms (one per monomer). A much more gross simplification may be achieved by noting that the protein backbone includes large stretches of characteristic local structure known as α-helices or β-strands. At low resolution, one is not interested in the detailed difference between a helix in protein 1 or protein 2, so comparison methods often treat these local or secondary structures as basic motifs.

Currently, a large number of different approaches exist for the detection of protein similarity.[4] The task in protein comparison is to find a representation which allows rapid location of characteristic units, regardless of their environment.[5] If one is content to work at the level of major secondary structure units, it may be possible to attempt something close to an exhaustive search of the similarities between the arrangement of such units.[6] Alternatively, one could move one protein over another, one monomer at a time, calculating some similarity measure at each step. This may provide a very non-optimal gapless alignment to serve as a seed alignment into which gaps could be introduced.[7] A more elaborate approach would be to calculate the intra-distance matrix for the C^{α} atoms of each protein, and then use a limited systematic approach to find one distance matrix in the other. This can again serve as the seed alignment for a refinement step, which would allow for gaps and chain breaks.[8]

Looking at Fig. 2, it may seem possible to try some dynamic programming approach to align the two strands. The difficulty is that aligning a part of protein 1 assumes one has already aligned the rest of protein 1 (since one wants to preserve intra-structural relationships). At some computational cost, it is possible to use a two-level (double) dynamic programming approach to find the optimum local alignment at each site. This is an approximation, but can be tuned to work well in practice.[9,10] The idea, however, suggests a different simplification which is pursued in this work.

If each protein could be represented as a characteristic string, the problem could be solved with a guaranteed optimal answer. Consider the strings shown in Fig. 3—clearly, an 'a' matches an 'a', and a 'd' matches a 'd'. This kind of string alignment problem can be solved using a dynamic programming method, in the most general case, in $O(n^3)$ time[11] or with a slight restriction in $O(n^2)$ time.[12] These string comparison methods are standard practice in protein sequence comparison, but comparing protein structures is a different problem. One could draw a protein as a linear chain, but this would not capture the three-dimensional aspect of the problem. One cannot compare an 'a' in one protein to a site in another protein. The structure does not depend on 'a', but on its relationship to the rest of the molecule. The analogy with sequence/string comparison does, however, suggest a tractable approach. If one had special symbols representing the patterns of interaction between sites of a protein, a matrix of site similarities could be constructed based on the result of the comparison of each symbol in the characteristic sets of two proteins. This matrix is essentially the same as one generated in the process of string alignment, thus dynamic programming can be used to find an optimum solution. If the function to measure the similarity between symbols can be defined, then one has a method to align any two protein structures, regardless of differences in size or the extent of the similarity.

A	A	b	c	-	b	a	d	q	R	s	a	d	c protein 1
A	A	b	x	a	b	a	d	-	-	-	a	b	c protein 2

Fig. 3. Alignment of two text strings.

One approach to defining such a symbol set is to consider chemical properties at each site. For example, each residue can be labelled with numbers quantifying the amount of side chain exposed to the solvent, the local secondary structure, or participation in hydrogen bonds.[13,14] In this work, we attempt a purely geometric approach, in the spirit of Gardiner et al.[15] First, each protein is converted into a graph representation where each vertex is a C^{α} atom. The edges between the vertices are labelled with the distance between atoms, but also with a set of four other numbers that characterize the local geometry and how the local geometry between the pair of C^{α} atoms is related. A first step of data reduction is applied by removing all edges exceeding a distance cutoff and locating each clique (fully connected subgraph) within each protein. Effectively, each vertex (site in a protein) is characterized by the cliques in which it participates, but it is the cliques themselves which form the characteristic set of symbols. The symbols are compared by determining their maximum common substructure and the abundance of other similar structure. We then find the longest, most similar regions of the two proteins by dynamic programming.

Graphical Representation of Molecular Structure

A protein fold is the characteristic three-dimensional (3D) shape defined by the path of the polypeptide through space and passing through the C^{α} of each residue. The local geometry at residue i is simply described by the unit vector tangent ($\vec{t_i}$) and normal ($\vec{n_i}$) of the path at the i^{th} residue C^{α}. If $\vec{p_i}$ is a vector from C^{α}_{i-1} to C^{α}_{i}, then:

$$\vec{t_i} = \frac{\vec{p_i} + \vec{p_{i+1}}}{\left|\vec{p_i} + \vec{p_{i+1}}\right|} \tag{1}$$

$$\vec{n_i} = \frac{\vec{p_{i+1}} - \vec{p_i}}{\left|\vec{p_{i+1}} - \vec{p_i}\right|} \tag{2}$$

The vectors at either end of the chain are defined as the corresponding adjacent tangents and normals. Each C^{α} site in the fold is characterized by pairwise contacts to other C^{α} near in space, within 10 Å. A pairwise contact between the i^{th} and j^{th} residue ($i<j$) is characterized by the contact distance d_{ij}, and the relative local backbone geometry between the sites. To define this, we take a unit vector from the coordinate of i and j:

$$\vec{e_{ij}} = \frac{\vec{r_j} - \vec{r_i}}{d_{ij}} \tag{3}$$

The relative orientation is then characterized by the products:

$$\begin{aligned} o_{ij,tf} &= \vec{t}_i.\vec{e}_{ij} \\ o_{ij,nf} &= \vec{n}_i.\vec{e}_{ij} \\ o_{ij,tb} &= -\vec{t}_j.\vec{e}_{ij} \\ o_{ij,nb} &= -\vec{n}_j.\vec{e}_{ij} \end{aligned} \quad (4)$$

where the subscripts *t* and *n* refer to products involving the tangent or normal vector and the vector of the contact, and *f* or *b* imply the direction (forwards or backwards) with respect to sequence ordering.

The structural features we have described are conveniently represented by a labelled, undirected graph. A graph **G(V,E)** is a pair of sets: the vertices, **V**, represent the set of C^{α} atoms and edges, **E**, are a set of unordered vertex pairs, which represent the network of pairwise contacts. Each pair (*ij*) is labelled with d_{ij} and the four scalars ($o_{ij,nf}$, $o_{ij,tf}$, $o_{ij,nb}$, $o_{ij,tb}$) from (4), and may appear only once in **E**.

We now define the basic requirement for structural comparison, a definition of similarity between two pairwise contacts, as represented by edges. The labels for the edge between *i* and *j* are collected into a set:

$$\mathbf{e}_{ij} = \{d_{ij}, o_{ij,nf}, o_{ij,tf}, o_{ij,tb}, o_{ij,nb}\} \quad (5)$$

Therefore, one can apply a boolean function $f_{\text{equiv}}(\mathbf{e}_{ij}, \mathbf{e}_{kl})$ to a pair of these label sets, using two arbitrary scalar tolerances D_{tol} and O_{tol}:

$$f_{eqiv}(\mathbf{e}_{ij}, \mathbf{e}_{kl}) = \begin{cases} 1 & \begin{array}{l} \left|o_{ij,tf} - o_{kl,tf}\right| \le O_{tol} \\ \left|o_{ij,nf} - o_{kl,nf}\right| \le O_{tol} \\ \left|o_{ij,tb} - o_{kl,tb}\right| \le O_{tol} \\ \left|o_{ij,nb} - o_{kl,nb}\right| \le O_{tol} \\ \left|d_{ij} - d_{kl}\right| \le D_{tol} \end{array} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Graphs, Subgraphs, and Common Subgraphs

A subgraph is a subset of vertices and edges contained in another graph. A subset of vertices is said to induce a subgraph, where the edge set is all edges covering the subset of vertices. A partial subgraph is the graph formed from a specified subset of the edges induced from a vertex set. If every vertex is connected to every other vertex, the graph is a clique. Cliques are graphs of maximal edge density, and not contained by any other larger clique.

If there exists a pairwise mapping between the vertex sets of two graphs which transforms a subset of the edges to some set of equivalent edges (under a criterion for label equivalence), the set of transformed vertices and conserved edges are called a common subgraph. The largest common subgraph is the maximum common subgraph (MCS). If the MCS contains all the edges of both graphs, then the graphs are isomorphic. There are often different mappings of the same subset of vertices which define the same common subgraph. The set formed by these distinct alternative mappings is the automorphism set of the subgraph, the set of vertex permutations that leave the edge set of a graph unchanged.

A Methodology for the Comparison of Two Protein Graphs

We have adapted an existing methodology, originally applied to object recognition for computer vision by Barrow and Burstall,[16] which transforms the MCS problem to one of finding cliques in a correspondence graph (CG). Fig. 4 demonstrates the CG derived from two-edge labelled, isomorphic graphs. The vertex set of the CG is the cartesian product of the vertex sets of the two parent graphs. A pair of parent nodes labels each node, so the vertex set contains all possible vertex equivalences. The edge set in the CG is constructed by connecting a pair of node mappings only if the edges between the mapped nodes in their respective graphs

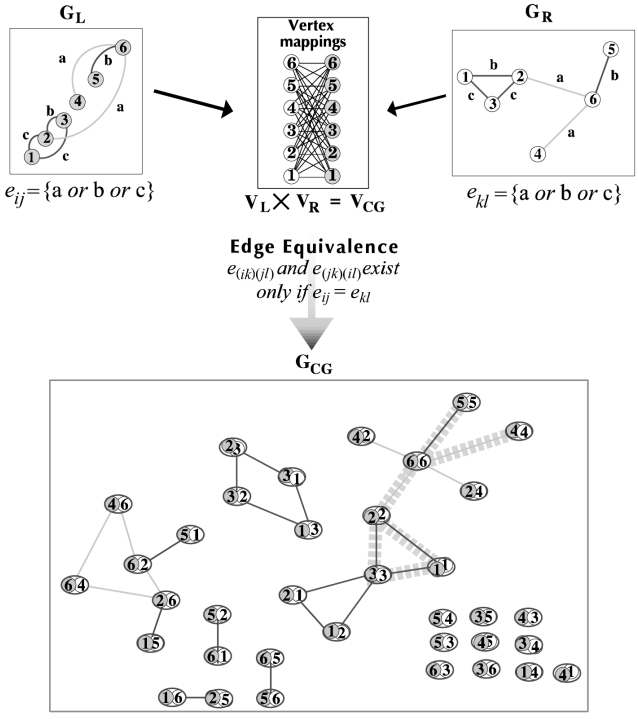

Fig. 4. A simple correspondence graph. In this example, $\mathbf{G_L}$ and $\mathbf{G_R}$ are edge-labelled (**a**, **b**, and **c**) isomorphic graphs. The graphs possess only one edge label preserving automorphism, and the MCS mapping corresponds the identically numbered vertices. $\mathbf{G_{CG}}$ is the CG. The subset of vertices in $\mathbf{G_{CG}}$, that map all the nodes of $\mathbf{G_L}$ and $\mathbf{G_R}$ correctly, induce a subgraph (outlined by grey dashes) isomorphic to the MCS. Because the MCS is connected, the mapping is part of the largest region of connected structure in $\mathbf{G_{CG}}$, which also contains the next largest subgraph isomorphism.

are considered to be similar. In this way, the nodes represent the search space of any MCS algorithm, and the edge connectivity represents the union of all common subgraphs. Traditionally, a clique enumeration algorithm is used to locate possible MCS fragments within the CG. The generation of a solution then requires the selection of the maximum subgraph mapping from the union of these cliques.

The boolean function (6) is used to select similar edges from protein graphs. This is a more relaxed label equivalence than the one in Fig. 4, so there are considerably more, locally similar, subgraph mappings which must be filtered. In this case, exhaustive enumeration quickly becomes impractical because the search space grows almost factorially. Some dramatic improvements can be gained from applying heuristic constraints to the search.[17] Alternatively, one could form a continuous optimization problem.[18] However, constraints are often based on an assumption of the particular form of the common subgraph, and any representation of the CG is still demanding to store and search for the comparison of very large, sparse graphs.

Because of this, a pragmatic heuristic has been employed. One of the parent (protein) graphs ($\mathbf{G_L}$), is broken up into its component cliques.[19] The CG can then be decomposed into corresponding 'slices', and more efficiently searched for the largest cliques only.[15,20] Finally, a scoring is applied to each clique in a slice of the CG, based on the density of edges and their geometric similarity, as defined in the next section. These scores are accumulated over the set of CG vertices for each clique. The summed scores for each CG vertex directly constitute an element $s_{ii'}$ of the similarity matrix **S**, and represent the similarity of site *i* in one protein, and site *i'* in the second protein. The dynamic programming algorithm[12] can then be directly applied to produce an alignment of the two original structures.

Clique Scoring

This comparison methodology is based upon recognition of the largest complete set of characterized contacts that are conserved in the local structure of a protein. However, these patterns are not unique — there is considerable redundancy in local monomer interactions. The set of maximal cliques ($\mathbf{M}_n$) found in any CG slice is an indistinguishable mixture of chance local similarities, and matches that are representative of globally similar features. If each maximal clique ($\mathbf{m}_{n'}$) is assumed to be independent, we may estimate the global significance of any matched edge in $\mathbf{M}_n$ by a ratio of observed abundance to a deductivean a priori expectation. The size of the clique, $|\mathbf{k}_n|$, which was used to form the CG slice, and the size of the set of automorphic mappings of the clique, $aut(\mathbf{k}_n)$ can provide a crude estimate:

$$\mathrm{p}(\mathbf{k}_{n'}\left|\mathbf{m}_{n'}\right|,\mathrm{M}_n)=\frac{\left|\mathbf{m}_{n'}\right|^2-\left|\mathbf{m}_{n'}\right|}{\left|\mathbf{k}_n\right|^2-\left|\mathbf{k}_n\right|}\cdot\frac{aut(\mathbf{k}_n)}{\left|\mathbf{M}_n\right|}, \qquad (7)$$

where we follow Bollobás[21] and use the convention that $|\mathbf{k}|$ refers to the number of vertices in **k**. The differences between the matched interactions provide further information to discriminate significant similarity. We simply measure this with the function $\mathrm{r}(e_{ij\ i'j'})$, a scaled sum of the deviation in local geometry for a pair of edge labels.

$$\mathrm{r}(e_{ij,i'j'})=4-\frac{3.\left|d_{ij}-d_{i'j'}\right|}{D_{tol}}-\frac{\left|o_{ij,nf}-o_{i'j',nf}\right|+\left|o_{ij,tf}-o_{i'j',tf}\right|+\left|o_{ij,nb}-o_{i'j',nb}\right|+\left|o_{ij,tb}-o_{i'j',tb}\right|}{O_{tol}} \qquad (8)$$

The product of (7) and (8) is accumulated in the similarity matrix for each vertex, over all maximal clique sets. In this way, we attempt to distinguish between the common local structure of the molecules, and the more significant long-range similarities that provide some biochemical insight.

Results and Discussion

The methodology should be fast enough for routine use, but is currently too computationally demanding for database searches. Some examples have been selected to show the performance of the approach. Comparisons were made with a D_{tol} of 2.5 Å and an O_{tol} of 0.7. All structures are drawn diagramatically with α-helices as thick ribbons and β-strands as thick arrows.

Fig. 1 shows a relatively simple case. Both proteins (1c9k chain B and 1sky chain B) are involved in triphosphate metabolism and a human expert may be able to use this information to predict a structural similarity. The complete structures are shown at the top of the figure, and one may see that the left-hand protein (1c9k) is, to low resolution, included in the right hand protein (1 sky). This is made clearer by the bottom pair of pictures with the common core extracted. It is clear that the β-strands and α-helices are generally well aligned. The figure also shows an important property of the methodology. Similarities between structural elements can be detected, even if there is some distortion of one molecule. This is important since an internal hinge motion may lead to large changes in long-range distances. This can be a problem for methods based on distance matrix comparison.

Fig. 5 shows a more interesting example based on 2fxb and 5fdl. Again, the top pictures show the complete proteins and the bottom shows the extracted common cores. These are both electron transport proteins with iron sulfide clusters, which may lead an expert to expect a similarity but with the complication that 5fdl has two metal clusters. In the bottom of the figure, one can see that the three large strands in the centre, as well as the small helix and strand at the bottom, are conserved. It is also clear that finding this alignment requires one large gap to be inserted.

This methodology can reveal more about overall similarities. Fig. 6 shows the similarity matrix for the two ferredoxins. The glyphs on the axes represent the secondary structure at each point along the sequences, and the black

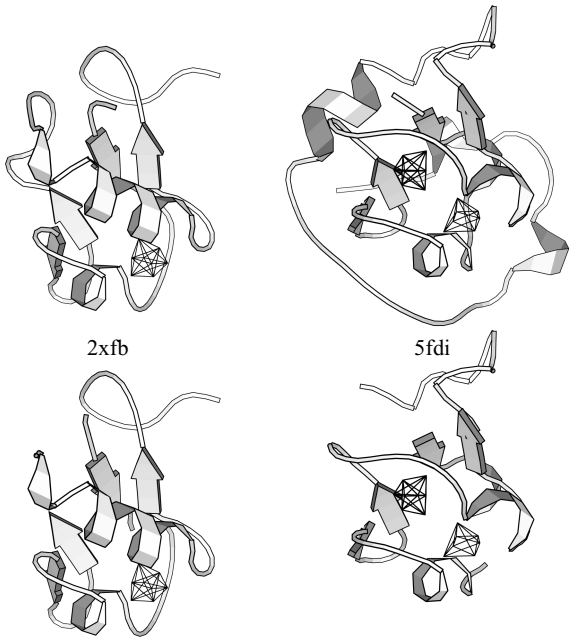

Fig. 5. Two ferrodoxins, 2fxb and 5fd1. The Fe_4S_4 and Fe_4S_3 clusters are shown as connected point sets. As in Fig. 1, the full structures are above the labels, while below are the common cores.

line running through the matrix from the lower left corresponds to the alignment in Fig. 5. Vertical and horizontal segments in the line correspond to extra structure in 2fxb and 5fdl, respectively. A typical feature of these matrices are the regions of high similarity that can be seen parallel with the optimal alignment. They correspond to fragments of alternative alignments between highly redundant secondary structure, which is common in many protein structure comparisons. However, the similarities of this pair of proteins have a more unusual property, seen by other workers.[14] If the sequence of one protein were cyclically permuted at the point marked by an arrow, the band of similarity to the right of the main band could be joined to the weaker diagonal lines towards the top left of the figure. Physically, this is the more correct alignment, corresponding all the Fe binding points between the two proteins, and both alignments match the repetitive structure in common between the two binding sites of 5fd1. In evolutionary terms, it may well reflect occurrence of events such as sequence or gene duplication.

The methodology for finding structural similarities between proteins works well, although it does take a rather abstract route. First the proteins are converted into graphs, then sets of cliques. The problem is then cast in terms of the

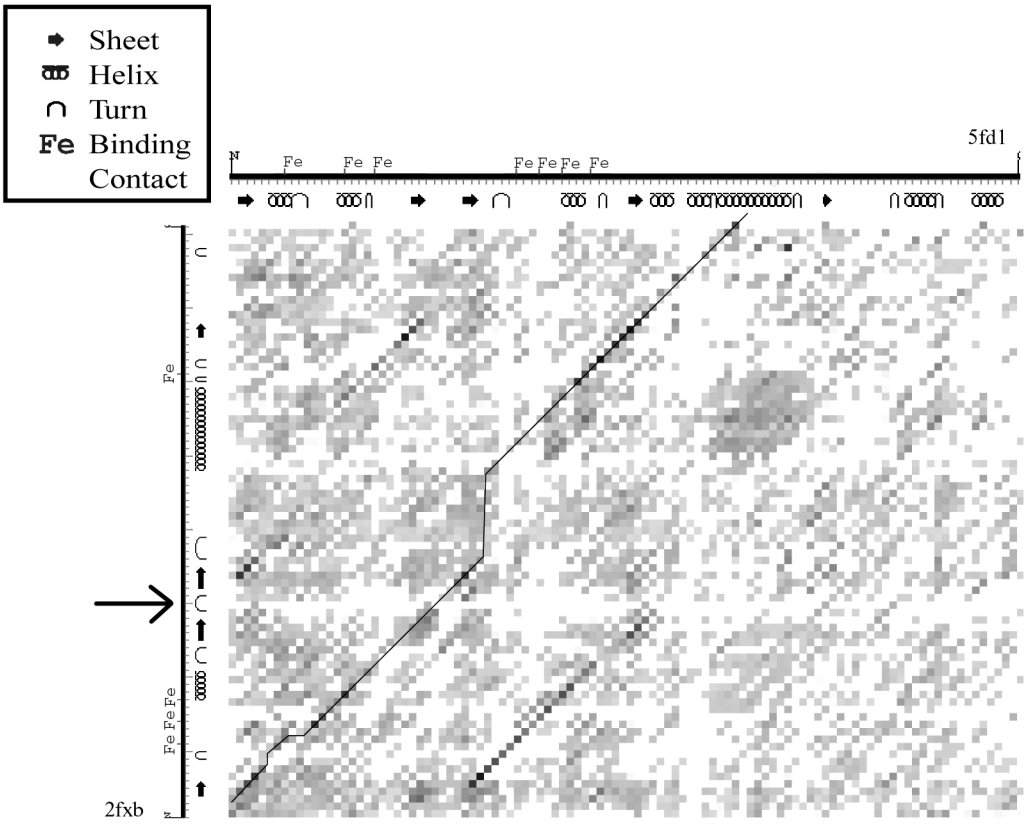

Fig. 6. Similarity matrix generated for 2fxb and 5fd1. The scales show secondary structure type along each sequence: helical (coil), β-strand (arrow), β-hairpin (loop), and the location of Fe binding sites generated with SEView.[22] The elements are shaded according to increasing vertex compatibility and the line through the matrix is the alignment in Fig. 5. The large arrow on the left shows the point where the cyclic permutation of 2fxb begins, resulting in an improved alignment of the residues involved in Fe_4S_4 binding.

maximal common subgraph problem, and finally transformed to allow the application of a dynamic programming algorithm. Essentially, we have adapted a non-sequential comparison approach, to make it practical for the comparison of larger and sparser graphs, by reducing the solution to a problem of selecting mappings from a matrix of similarity scores that is generated between all pairs of vertices from two graphs.

Our approximation usually gives an appropriate weight to similar sites in the protein, although along the path of a solution it can be seen that the weighting is not consistently high. Sometimes, clearly similar stretches of backbone have little or no similarity score. The latter must mean that the pruning criterion is too stringent for some types of fold similarities, due to cliques formed by sets of local interactions not always being completely conserved between structurally similar proteins. This is particularly true for distant evolutionary homologues—where the local backbone geometry is not well conserved, but the overall shapes of the molecules are quite similar. Such a resemblance can be discovered with a lower resolution characterization of the molecule, where the arrangement of more than three residues is taken into account.

The results show that there are some arbitrary thresholds and the method can be tuned to tolerate more or less similar regions as one desires. Stronger similarities may be required for collecting reliable parameterization data, while weaker similarities may be of interest in interpreting other calculations or looking for remote biological properties.

Using a graphical representation for structural characterization allows us to easily extend the formalism. Lower resolution characterization could be achieved, or we may include other types of interaction data, such as residue side-chain interactions, and solvent accessibility. Recently, Gardiner et al.[23] applied enumerative searching to graphs that explicitly represent protein surface hydrogen bonding sites, and found that complementary topologies can be identified which are involved in protein–protein interactions.

The limiting factor of our decomposition approach is the density of the graph formed in the representation. Because we decompose the search at the level of cliques, it would be quite inefficient to compare large, fully connected graphs because of the growth properties of the maximal clique problem. The solution found would also be much more sparse, because possible solutions are limited to the set involved in the maximal consistent cliques, which may preclude many mappings which are part of the MCS. However, the short-range spatial interpretation does seem to characterize structural similarity.

A final consideration is the application of dynamic programming. This method produces classically accepted alignments and has the usually desirable property that it will find the longest alignments consistent with the order of residues in each protein. However, if one is interested in finding features such as the possible cyclic permutation of the ferredoxin proteins, one could apply appropriate heuristics, but at a computational cost. Furthermore, the methodological extensions, which allow a search for less sequential similarities, must necessarily cope with the problems involved in direct search. Many heuristics, in the form of assumptions and thresholds, can be applied to overcome these, but they can only select specific forms of structural resemblance. It is therefore no surprise that different protein comparison methods continue to appear and provide different results.

Conclusions

We have presented one technique for limiting the space of spatial similarities that must be searched in order to identify protein structural resemblance. This involved a graph-theoretic transformation that selects particular mappings, and a numeric transformation allowing the simplification of the problem. It may be that the strict, local maximum, common subgraph property is too stringent for this particular graphical representation of protein structure, but the results presented here are sufficient to show its effectiveness. Finally, given the general nature of the method, the results suggest that the technique should find wide application in areas quite removed from protein structure comparison, but where the entities, which are to be compared, can be represented as graphs.

Acknowledgment

We thank Dr A. Russell for the implementation of the Gotoh algorithm used in this work.

References

[1] C. Hadley, D. T. Jones, *Structure* **2000**, *7*, 1099.
[2] J. Moult, T. Hubbard, K. Fidelis, J. T. Pedersen, *Proteins* **1999**, *S3*, 2.
[3] P. Kraulis, *J. Appl. Crystallogr.* **1991**, *24*, 946.
[4] E. Eidhammer, I. Jonassen, W. R. Taylor, *Reports in Informatics* **1999** No. 74 (Department of Informatics, University of Bergen).
[5] O. Bachar, D. Fischer, R. Nussinov, H. Wolfson, *Protein Eng.* **1993**, *6*, 279.
[6] J.-F. Gibrat, T. Madej, S. H. Bryant, *Curr. Opin. Struct. Biol.* **1996**, *6*, 377.
[7] R. B. Russell, G. J. Barton, *Proteins* **1992**, *14*, 309.
[8] L. Holm, C. Sander, *J. Mol. Biol.* **1993**, *233*, 123.
[9] W. R. Taylor, C. A. Orengo, *J. Mol. Biol.* **1989**, *208*, 1.
[10] C. A. Orengo, W. R. Taylor, *J. Theor. Biol.* **1990**, *147*, 517.
[11] S. B. Needleman, C. D. Wunsch, *J. Mol. Biol.* **1970**, *48*, 443.
[12] O. Gotoh, *J. Mol. Biol.* **1982**,*162*, 705.
[13] M. Suyama, Y. Matsuo, K. Nishikawa, *J. Mol. Evol*. **1997**, *44*, S163.
[14] J. Jung, B. Lee, *Protein Eng*. **2000**, *13*, 535.
[15] E. J. Gardiner, P. J. Artymiuk, P. Willett, *J. Mol. Graph.* **1997**, *15*, 245.
[16] H. G. Barrow, R. M. Burstall, *Inf. Process. Lett.* **1976**, *4*, 83.
[17] I. Koch, *Theor. Comput. Sci.* **2001**, 250 1.
[18] K. Mizuguchi, N. Gō, *Protein Eng.* **1995**, *8*, 353.
[19] C. Bron, J. Kerbosch, *Comm. ACM* **1973**, *16*, 575.
[20] R. Carraghan, P. M. Pardalos, *Oper. Res. Lett*. **1990**, *9*, 375.
[21] B. Bollobás, *Extremal Graph Theory* **1978** (Academic Press: London).
[22] T. Junier, P. Bucher, *In Silico Biol.* **1998**, *1*, 13.
[23] E. J. Gardiner, P. Willett, *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 273.