Chapter 2
Comparison of protein three-dimensional structures

Mark S. Johnson and Jukka V. Lehtonen
Department of Biochemistry and Pharmacy, Åbo Akademi University, Tykistökatu 6 A, 20520 Turku, Finland.

1 Introduction

In this chapter we define the different types of questions that may be asked through the comparison of the three-dimensional (3-D) structures of proteins, how to make the comparisons necessary to answer each question, and how to interpret them. We shall focus on the different strategies used, and the assumptions made within typical computer programs that are available.

Protein structure comparisons are often used to highlight the similarities and differences among related—*homologous*—3-D structures. Homologous proteins are descended from a common ancestral protein, but have subsequently duplicated, evolved along separate paths, and thus changed over time. The independent evolution of related proteins with the same function, *orthologous* proteins, which are found in different species, and the *paralogous* proteins, which have evolved different functions, all retain information on the original relationship. The amino acid sequences change over time reflecting the mutations, insertions and deletions that occur in their genes during evolution, and for many proteins the sequences themselves are so similar that common ancestry is apparent. For others, the sequences can be so dissimilar that the case for *homology* may be difficult to make on the basis of the primary structure. Nonetheless, comparing the 3-D structures when they are available can identify homologous proteins. This is possible since the evolution of proteins occurs such that their folds are highly conserved even though the sequences that encode them may not be recognizably similar.

Homologous proteins are often compared in order to highlight features (typically the amino acids and their relative orientations to one another), which have come under strong evolutionary pressure not to change because of structural and functional restraints placed on them. Conversely, differences in an otherwise conserved active site or binding site are used to explain differences in observed function.

Dayhoff and coworkers (1) long ago predicted that about 1000 different protein families should exist in nature, and it has become clear over recent years that most newly-solved 3-D structures do fall into an existing family of structures (2, 3). The approximately 100 000 proteins encoded in the human genome, whose sequences will be known early in this century, will fall within this limited number of families. Thus, one key bioinformational goal has been to compare and classify all proteins and their component domains into family groups, and one immediate goal is to solve at least one representative structure for each sequence family that is not obviously connected to any existing structural family. This single representative structure can then be used in knowledge-based modelling (4) to estimate the 3-D structures for other members of the family.

Comparisons are also made among non-homologous proteins to try and highlight structural features that are locally similar, but whose present-day sequences have not arisen as a consequence of evolutionary divergence from a common ancestor. Classic examples include the active site similarities among serine proteinases, subtilisins and serine carboxypeptidase II (5), each of which invoke the participation of histidine, serine and an aspartic acid in their proteolytic mechanism of action. The folds are different and the relative positions of these key amino acids along the sequence are different too. In the 3-D structures, however, the residues are similarly positioned to reproduce a common catalytic mechanism that has been exploited by nature on at least three separate occasions. Comparisons among non-homologous proteins can highlight structural units that are common features of the protein fold and comparisons have been made to classify amino acid conformations, regular elements of secondary structure (helices, strands, turns), supersecondary structure, and cofactor and ligand binding sites.

The comparison of protein structures can be achieved in many different ways. In this chapter, we present several of the basic procedures used in the wide variety of programs that have been developed over the years. These methods range from rigid-body comparisons, to methods more typical of sequence comparisons—dynamic programming, and to those methods that employ Monte Carlo simulations, simulated annealing and genetic algorithms to find solutions for combinatorially-complex structural comparison problems. We will describe methods that demand partial solutions as input to the procedure, as well as strategies for automatic hands-off solutions; and approaches to both homologous and non-homologous structural comparisons.

2 The comparison of protein structures

2.1 General considerations

The optimal superposition of two identical 3-D objects can be determined exactly. This only requires the calculation of (a) a translation vector to place one copy of the object over the other at the origin of the co-ordinate system and (b) a rotation matrix that describes the rotations needed to exactly match the two copies of the object. The *translation vector* describes movements along the x, y, and z directions in the co-ordinate system. The *rotation matrix* describes the α, β, and γ rotations in the three orthogonal planes. One of the main tasks of many super-positioning procedures is to define these values and then to apply them to the co-ordinates of the objects and they will then be superposed on each other. Two identical objects will have all points superposed exactly.

The major difficulty with non-identical objects, such as a pair of protein structures, is that they typically have different numbers of amino acids, different amino acids with different numbers, types and connectivities of atoms. Furthermore, amino acids present in one structure can be missing in the other: insertions and deletions—the gaps seen in a sequence alignment. Thus, except in the case of one protein co-ordinate set being compared with itself, no two proteins will have atoms in exactly identical positions. A protein whose structure has been solved several times will also vary with overall differences in the main chain co-ordinates of no more than about 0.3 Å, but they will be different.

The superposition of most protein structures as rigid-bodies, therefore, is not straightforward, and several different considerations need to be resolved in advance of the comparison. These include:

(a) Which atoms will be compared between the molecules?

(b) How will the dissimilarity or similarity between relative positions of matched atoms be taken into account?

(c) Should the structures be compared as rigid-bodies (in most cases, resulting in a partial alignment of the most similar regions, which can be displayed graphically)? Or have significant structural shifts occurred that require a procedure that can accommodate these changes (typically providing the complete alignment of the sequences, including gap regions, on the basis of the structural features compared)?

(d) How will one define what constitutes an equivalent matched set of co-ordinates between non-identical objects where exact matching of atoms will only rarely be seen?

(e) How will the program be initially seeded? Many methods need to be supplied with co-ordinates of a set of equivalent atoms at the onset of the comparison, a minimum of three matched pairs, thus requiring some information on the likely superposition of the two structures in advance of comparison.

(f) How will the quality of the structural comparison that results be assessed?

2.2 What atoms/features of protein structure to compare?

Depending on what question you wish to answer by comparing a pair of structures, the choice of which atoms' co-ordinates will be superposed can be crucial. For example, to look at similarities/differences surrounding a bound cofactor common to two proteins, you may choose to superpose all or some of the atoms of the cofactor, apply the translation vector and rotation matrix to the entire co-ordinate file—protein and cofactor included. Alternatively, the backbone co-ordinates of the proteins could be superposed and the relative positions of the cofactors examined after the superposition.

It is usually not very useful to compare atoms of amino acid side chains when making global structural comparisons. Different amino acid types have different number of atoms and different connectivities that can preclude their direct comparison. Residues, even identical ones, will have different conformations, especially when they are located at the solvent-exposed surface of the proteins. However, there are situations where the local comparison of side chains can be very useful, for example, in the comparison of residues lining an active or binding site especially when different ligands are bound to the same or similar structures.

For most general methods, which aim to superimpose two proteins over the maximum number of residues, the Cα-atom co-ordinates are typically employed (all atoms of the protein backbone and even the side chain Cβ-atom, but excluding the more positionally-variable carbonyl O, can also be used). (Except where noted, we will consider Cα-atom co-ordinates in the protocols described herein.) Whereas the side chain conformations can vary wildly between matched positions in two structures, the Cα-atom or backbone trace of the fold is typically well conserved, with regular elements of secondary structure, the α-helices and

Table 1 Examples of features[a] of proteins that can be used in comparisons

Properties	
(a) Residues	**(b) Segments**
Identity	Secondary structure type
Physical properties	Amphipathicity
Local conformation	Improper dihedral angle
Distance from gravity centre	Distance from gravity centre
Number of neighbours in vicinity	Average C$^{\alpha}$ density
Position in space	Position in space
Global direction in space	Global direction
Main chain accessibility	Main chain accessibility
Side chain accessibility	Side chain accessibility
Main chain orientation	Orientation relative to gravity centre
Side chain orientation	
Main chain dihedral angles	
Relations	
(a)	**(b)**
Disulfide bond	Relative orientation of two or more segments
Vectors[b] to one or more nearest neighbours	Vectors[b] to one or more nearest neighbours
Distances to one or more nearest neighbours (e.g. atom pairs or contact maps)	Distances to one or more nearest neighbours
Change in number of neighbours in vicinity	
Ionic bond	
Hydrogen bond	
Hydrophobic cluster	

[a] See refs 7, 8, 10, 11.

[b] Vector defines both distance and direction in the local reference frame.

β-strands, matching closely and sequentially along the fold of the two structures. Differences in Cα-atom traces are more often seen at loop regions that connect the strands and helices in proteins: Frequently these loop regions are exposed to the solvent at the surface of the protein and thus have fewer constraints placed on their conformations.

For more dissimilar protein structures, rigid body movements and other structural changes can occur in one structure relative to the other. When this happens, rigid-body comparisons of the 3-D structures can often lead to poorly matched structures, although the folds are the same. If these changes are not large, then dynamic programming procedures (6) that consider only Cα-Cα atom distances or other structural properties of the amino acids (*Table 1*) after an initial rigid-body comparison can be quite effective in matching all residues from the protein structures (7–9). Others have described automated procedures that involve the comparison of structural relationships that require special techniques to solve these problems of combinatorial complexity (7, 8, 10, 11).

Protocol 1

Features used for the comparison of protein 3-D structures

Distances between atomic co-ordinates are often used (a) for more similar proteins where rigid-body shifts of one structure relative to another are not a significant factor, (b) to illustrate the degree of structural change, or (c) where a local comparison of a site of interest—active site or binding site—is desired for visualization purposes. Where significant changes to the structures have occurred, other structural features, which are not as sensitive to these relative structural shifts, can be compared in addition to atomic co-ordinates.

Rigid-body structural comparisons

1 Choose the atoms for comparison that are appropriate for the question to be asked. Most often, but not necessarily, the Cα-atom co-ordinates are used by default.

2 Comparisons will then be based on the distances between atoms that are considered to be equivalent. For rigid-body methods, a distance cut-off is used to define equivalent matched positions. Typically, the cut-off value is on the order of 3 Å, although values between 2.5 Å and 4.5 Å have been used. Lower values are more restrictive and will lead to fewer aligned positions in more dissimilar structures.

Structural feature comparisons

1 Features of individual atoms, residues or segments of residues, both properties of and relationships between individual atoms, residues or segments, are considered either separately or in combination with each other as a basis for structural comparisons (see *Table 1* and refs 7, 8).

2 Comparisons will be based on differences/similarities between potential matched regions in the two structures in terms of the features compared. An alignment

Protocol 1 continued

algorithm is used to give the best 'sequence alignment' based on the structural features that have been supplied.

(a) Property comparisons may require an initial alignment (e.g. rigid-body).

(b) Relationships can be aligned by a variety of methods, e.g. Monte Carlo simulations (11), simulated annealing (7), double dynamic programming (10), genetic algorithms.

3 The structures can subsequently be superposed according to the matches in the alignment, but a single global superposition may be meaningless when large movements, such as domain movements, have taken place. In that case, each domain should be superposed separately.

Dynamic programming methods can align structures on the basis of differences/similarities between any number and combination of *properties*—which are features of individual residues or segments of residues contiguous in sequence. In order to compute the difference or similarity between positions in a structure, for example on the basis of Cα–Cα distances, an alignment is required to give an estimate of the distances between atoms in the structures. Other structural properties, such as residue solvent accessibility, can be used with dynamic programming directly, but may provide less useful information for the comparison. *Relationships*—features of multiple non-sequential residues (*Table 1*): e.g. patterns of hydrogen bonding, hydrophobic clusters, Cα-atom contact maps—can also be compared. Monte Carlo simulations (11), simulated annealing (7), and double dynamic programming (10) have all been used to equivalence relationships among residue sets from structures. Each of these methods gives an alignment of the structures in the form of a sequence alignment, but to visualize the results of the comparison, a rigid-body comparison would still be required. This could be made over all matched positions or over those positions that matched 'best' according to the comparison criteria used. The global rigid-body superposition based on the alignment may also be unsatisfactory if large structural changes have taken place. To accommodate very large changes, such as domain movements, the domains can be superposed separately.

2.3 Standard methods for finding the translation vector and rotation matrix

For methods that compare the relative atomic positions in two structures, A and B, and produce the superposed co-ordinates as output, it is necessary to determine a translation vector and the rotation matrix that, when applied to the original co-ordinates, will generate the new co-ordinates for the superposed proteins. Firstly, the centre-of-mass of the each protein is translated to the origin of the co-ordinate system. Secondly, one of the structures is rotated about the three orthogonal axes in order to achieve the optimal superposition upon the other structure. Because the atoms chosen for comparison will not match exactly in terms of their relative atomic positions after superposition, a least-squares method is typically used to achieve the optimal superposition.

The following function minimizes the residual δ, which is expressed mathematically as:

$$\delta = \sum_{i=1}^{N} w_i \, (\overset{\varpi}{A}_i^{eq} - \Re\overset{\varpi}{B}_i^{eq})^2$$

where $\Re$ is the rotation matrix being sought that minimizes the differences between a total of N equivalent co-ordinate sets $\overset{\omega}{A}^{eq}$ from the first protein and $\overset{\omega}{B}^{eq}$ from the second protein; *wi* is a weighting that can be applied to each *i*th pair of equivalent positions.

Numerous methods have been developed to solve this pairwise least-squares problem in a variety of different ways (12–16). Others have described more general methods suitable for the least-squares comparison of more than two three-dimensional structures (17, 18) In our experience, the method of Kearsley (19) is a straightforward and simple means to obtain the optimal rotation matrix for a set of equivalent co-ordinates. We will only consider this procedure here (*Protocol 2*).

The major obstacle to solving the least-squares problem is that matched atom pairs from the two structures to be compared need to be specified to the algorithm at the beginning of any calculations. Thus, the computer program requires some idea of the final alignment before it can proceed. There are common situations where the comparisons would be made over a pre-defined set of residues: for example, (a) comparisons over residues that line an active site or binding site—to highlight similarities and differences over those positions; (b) comparisons of independent structure solutions for the same protein. In these cases, the atomic positions to be compared are usually known *a priori*, and a single round of rigid-body comparison is sufficient to obtain the optimal match. Frequently, however, global comparisons are made between proteins where the best-matched positions are not obvious in advance. In the case of similar protein structures, the requirement of an initial set of matches to seed the comparison is inconvenient at best, requires the pre-analysis of the proteins involved, and in the case of more dissimilar proteins, may be difficult to define. Additionally, we have often observed that when part of the answer is specified at the beginning of the comparison, then the final solution can be prejudiced to give a final result that is not necessarily the optimal one: The comparison was locked into a set of possible solutions by the information supplied to seed the procedure. Despite these criticisms, there are many good methods that employ this strategy.

For example, Sutcliffe *et al.* (16) specify a set of at least 3 Cα-atoms common to the two structures (3 positions define a unique plane in each structure). Good candidates for these common residues, supplied *a priori*, can be conserved residues at an active site or ligand binding site, be positions conserved in terms of the sequence similarity, or can be equivalent positions observed to form part of the common fold when the proteins are examined on a graphics device. This and other similar methods use an iterative procedure to progress towards better and better solutions that incorporate more and more equivalent atom pairs. (Later in this chapter we will detail several automatic strategies that have been used to get around this need for predetermining a set of equivalent atoms at the onset of the structural comparison.)

In the equation describing the residual (above), $\vec{A}^{eq}$ and $\vec{B}^{eq}$ contain the *x*, *y*, and *z* axes co-ordinates for exactly the same number of atoms from each of the two structures. These atoms are termed *equivalent* positions, and are those aligned positions that the superposition will now be calculated for. All other atoms in the molecules are ignored in determining the superposition, but the translation vector and the rotation matrix determined on the basis of these equivalent positions is subsequently applied to all atoms in the co-ordinate file, including any bound ligand, metal ions, and water molecules. Here, we will detail how to calculate the translation vector for each protein and describe one simple yet elegant method for determining the rotation matrix, developed by Kearsley (19, 20), which we use as the method of choice for our own procedures (*Protocol 2*).

Protocol 2

Rigid-body structural comparisons: translations and rotations

This protocol details the steps required to optimally superpose the equivalent atom co-ordinates from two proteins.

Data required

The co-ordinates of all atoms in the proteins' co-ordinate file (minimum of the Cα-atom co-ordinates) and the matched equivalent atoms in the two proteins.

The translation vector

1 Calculate the centre of mass from the *x*, *y*, and *z* co-ordinates for each set of equivalent atoms $\vec{A}^{eq}$ and $\vec{B}^{eq}$ from the two structures. For *N* atoms in the equivalent set of the first protein:

$$\vec{T}_A = \sum_{i=1}^{N} \vec{A}_i^{eq}/N$$

In other words, sum all of the *x* co-ordinates together and divide by *N* to give the average *x* co-ordinate for the equivalent set of atoms; repeat for the *y* and *z* co-ordinates. Repeat for the corresponding *N* equivalent atoms in the second structure:

$$\vec{T}_B = \sum_{i=1}^{N} \vec{B}_i^{eq}/N$$

Thus, the centre of mass is a single *x*, *y*, and *z* co-ordinate set for each of the proteins.

2 Translate both structures, all atoms in the file, so that their centres of mass (according to the set of equivalent atoms used) are located at the origin of the co-ordinate system. For *every* atom *i* in the first structure:

$$\vec{A}_i^{all}\,(trans.) = \vec{A}_i^{all}\,(old) - \vec{T}_A.$$

Protocol 2 continued

In other words, subtract the *x*, *y*, and *z* co-ordinate values for the centre of mass from the *x*, *y*, and *z* co-ordinate values for every atom in the co-ordinate file. Repeat for the second structure:

$$\overset{\omega}{B}_i^{all}\,(trans.) = \overset{\omega}{B}_i^{all}\,(old) - \overset{\varpi}{T}_B.$$

The rotation matrix: the Kearsley method (ref. 19) minimizes the average difference between sets of atoms using quaternion algebra

1. Generate a symmetric 4 × 4 matrix by adding selected combinations of differences and sums of co-ordinates calculated for each matched pair of equivalent atoms to the elements of the matrix (19). These are the co-centred co-ordinates, but only the co-ordinates of equivalent matched atom pairs, $\overset{\omega}{A}_i^{eq}$ and $\overset{\omega}{B}_i^{eq}$ are used at this stage.
2. Diagonalize the 4 × 4 matrix in order to obtain its eigenvalues and eigenvectors (see ref. 21 for general procedures).
3. Select the lowest eigenvalue and use elements of the corresponding eigenvector to construct the 3 × 3 rotation matrix $\Re$ (see ref. 19 for details).
4. Multiplication of each co-ordinate in the second structure B by $\Re$ will produce the superposition of the entire structure onto protein A, where the average distance between matched atoms of the equivalent set is a minimum: $\overset{\omega}{B}_i^{all}\,(trans.,rot.) = \Re \times \overset{\omega}{B}_i^{all}\,(trans.)$.
5. The selected eigenvalue divided by the number of atom pairs in the equivalent set is equal to the square of the RMSD after rotation. $\Re$, calculated above, leads to the superposition whose RMSD is a minimum for these sets of equivalent atoms.

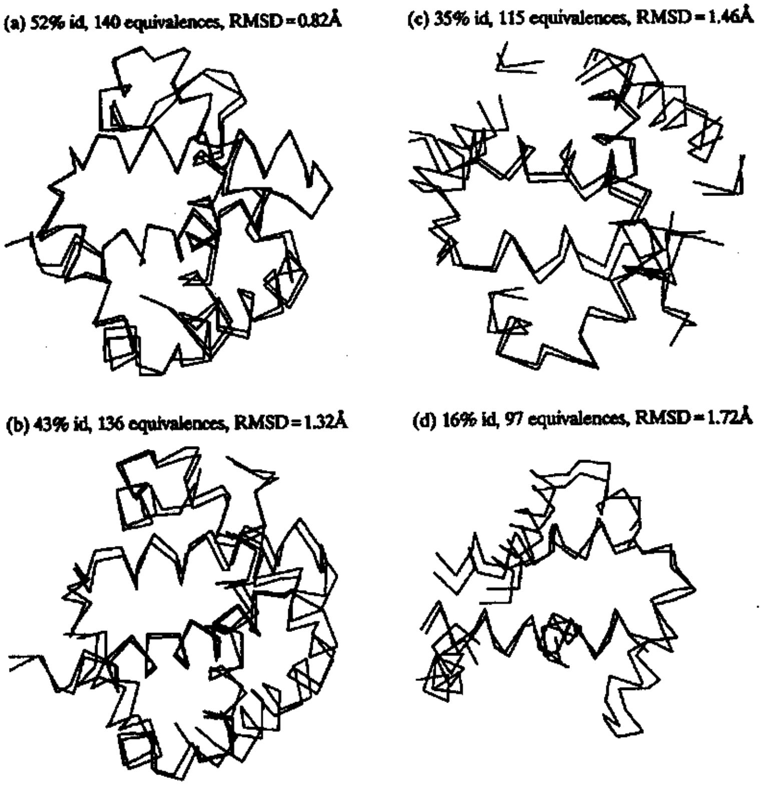
2.3.1 Structural alignment of sequences

In *Figure 1*, is shown the loss of superposed Cα-atoms in globin comparisons as the percentage sequence identity decreases. As an alternative to rigid-body structural comparisons, especially when the rigid-body structural similarity is reduced due to modest structural alterations, other methods have been developed that provide the alignment of the sequences of the structures. Nonetheless, rigid-body comparisons are often used in combination with these other procedures or for visualisation of the results.

For example, the dynamic programming algorithm described below can make comparisons on the basis of Cα–Cα atom distances, as well as other features (see *Table 1*).

(a) As we have stated above, a rigid-body comparison is often needed in order to make comparisons of structural properties suitable for dynamic programming alignment.

(b) The dynamic programming method is often used in conjunction with rigid-body super-positioning methods in order to efficiently assign equivalent matches.



(a) 52% id, 140 equivalences, RMSD = 0.82Å

(c) 35% id, 115 equivalences, RMSD = 1.46Å

(b) 43% id, 136 equivalences, RMSD = 1.32Å

(d) 16% id, 97 equivalences, RMSD = 1.72Å

Figure 1 Reduction in the extent of the common equivalent matches in pairwise structural superpositions as a function of decreasing percentage sequence identity. Traces of the backbones are shown for Cα-positions within 2.5 Å after rigid-body superposition with the computer program MNYFIT (16). The haemoglobin α-chain of *Pagothenia bernacchii* (Protein Data Bank (PDB, ref. 51) code: 1PBX) is aligned in (a) with the α-chain of equine haemoglobin (2MHB) and in (b) with the β-chain of human haemoglobin (2HHB). (c) The human haemoglobin β-chain (2HHB) aligned with the sea lamprey globin (2LHB). (d) The erythrocruorin of *Chironomous thummi thummi* (1ECD) aligned with the leghaemoglobin of *Lupinus luteum* (1LH1). (From ref. 4, with permission.)

(c) The dynamic programming algorithm produces a full alignment of all positions in the structures (residues are aligned with each other or with gaps), while the rigid-body methods align fewer and fewer potions in the structures as the sequence similarity decreases (*Figure* 1).

(d) Dynamic programming algorithms do not give a superposition of the structures suitable for visualization. This can be obtained from the alignment by applying the rigid body method to the defined matched pairs.

(e) Dynamic programming can often lead to alignments of the structures where rigid-body movements have occurred in the structures themselves. For example, the large movements of the entire domains seen in the liganded and unliganded structures of the periplasmic bacterial lysine–arginine–ornithine binding protein (*Figure* 2). Rigid-body comparisons can be applied,

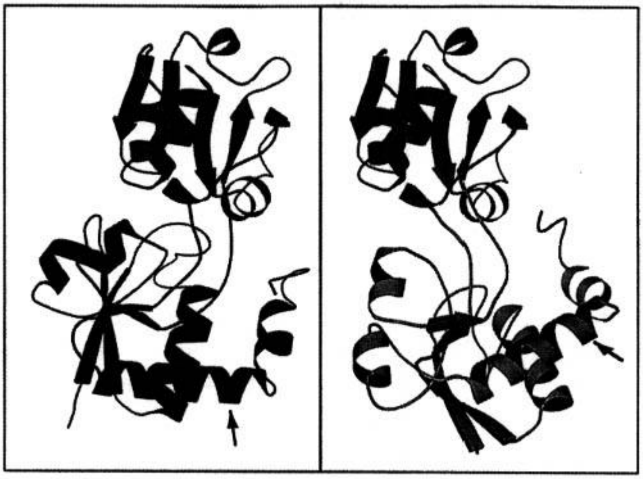


Figure 2 Two different conformations of the 3-D structure for the same protein, the lysine–arginine–ornithine binding protein from *Salmonella typhimurium*. *Left*: the structure of the protein in complex with lysine (1LST), lysine not shown. *Right*: the uncomplexed structure (2LAO). The smaller domain on the upper part of the figures is in same orientation and the arrow pointing to the Cα-atom of Glu 216 illustrates the magnitude of the movement of the larger domain at the bottom of the figure. Figure prepared with MOLSCRIPT (52).

however, to the domains separately to pinpoint any changes within each domain that have occurred upon ligand binding; while superpositioning on one domain can be used to highlight the relative movements that have occurred between the domains upon binding.

2.4 Standard methods to determine equivalent matched atoms between structures

There is no exact definition of topological equivalence, and the criteria used can vary from method to method. In rigid-body superposition methods, a distance cut-off between equivalent atoms is frequently used. In methods were other structural features are considered, all aligned positions might be considered to be topologically equivalent between two structures, or they may be assigned according to the degree of positional similarity of features used to make the comparison.

2.4.1 Definitions of structural equivalence: the alignment

In determining a set of equivalent atom sets, distance criteria are often used. After one structure has been superposed on another, topological equivalent atoms can be limited to those atom types under consideration that are within a distance cut-off value. The Euclidean distance, D, between two points is:

$$D_i = \sqrt{(x_{A(i,eq)} - x_{B(i,eq)})^2 + (y_{A(i,eq)} - y_{B(i,eq)})^2 + (z_{A(i,eq)} - z_{B(i,eq)})^2}.$$

In rigid body comparisons, where the Cα-atoms of the protein backbone have been used as a basis for comparison, a distance cut-off typically in the range 2.5 Å to 4.5 Å has been used. Values above 3 Å lead to more multiple matches to a single atom: the distance between two consecutive Cα-atoms along the protein backbone is around 3.5 Å. Lower values will reduce the number of equivalent matches when more dissimilar proteins are compared. Distances or dissimilarity measures will also be required for the comparison of other structural features, both properties and relationships, see ref. 7 for example. Common to both rigid-body methods, which rely on simple distance data, and other methods, which incorporate other types of information into the alignment process, is the need to determine the matching of locations between the structures to be superposed (*Protocol 3*). This can be part of an iterative procedure to provide a new set of equivalent atoms that are then used to determine a new translation vector and rotation matrix in order to improve a match. This is also one of the final steps in any comparison procedure, where the resultant alignment is determined. Three basic approaches have been used: (a) dynamic programming, (b) graph theoretical match list handling and clique detection methods, and (c) methods more suitable for solving combinatorially-complex matching problems.

The Needleman and Wunsch (6) method is a convenient fast method for aligning proteins. By scoring all possible pairs of matches between two structures, the method insures that the optimal scoring solution is found for the scoring scheme employed. The method accommodates a loss of elements in one structure relative to another—the gaps corresponding to insertions and deletions. Thus, the method provides a full alignment where every residue position in each protein is matched to either a residue position in the other protein or a gap. Thus, this method can efficiently resolve the multiple matching and many combinatorial problems seen with the list sorting procedure. Once structural relationships have been equivalenced between a pair of structures, this information can also be used within the dynamic programming method.

With the match list sorting procedure, for example ref. 22, possible equivalent matches between the proteins are tabulated: matches of protein B to each position in protein A in one list, and matches of protein A to protein B in a second list. These lists contain both authentic matches of conserved structure, chance matches that need to be eliminated from the lists and multiple matches between one element in one protein to several different elements in the other protein. The challenge, then, is to cull these lists by keeping the best matches (i.e. matches that can extend a series of previous matches, have a good matching score or give a good fit), removing structurally unlikely matches (matches that are not co-linear—are out of sequence with other matches—and isolated matches that do not extend further other matches), and by reducing multiple matches to single matches.

A more elaborate approach was introduced by Mitchell *et al.* (23). Their method does not filter out extraneous matches, but instead tests each combination of matches to find the optimal equivalent set. As a result, a 'clique', the maximum sub-graph common to two graphs representing the structures is found. The clique detection algorithm is based on graph theory and offers a way to find similar parts of structures that have not been superimposed. The basic idea is to represent each structure as a graph of nodes and vertices. Each node corresponds to either an atom, piece of main chain, secondary structure element, or similar definite piece of structure. Each vertex is a relation between two nodes in a structure: the distance between the atoms, vector from one atom to another (both distance and direction in some co-ordinate frame), distance and angle between two secondary structure elements, or more a more complicated distance measure involving other properties of the nodes. If two structures contain a similar substructure, then the nodes belonging to that substructure are connected in both structures by very similar vertices. The task is to find the maximal common sub-graph from the set of possible common sub-graphs. While this is a NP-complete task, it is feasible due (a) efficient search algorithms evolved within graph theory, and (b) the use of (few) secondary structure elements (SSEs) as the compared pieces of the structures instead of (many) atoms. Several other programs have been described that use a very similar approach (see ref. 24 and citations therein); the main differences are in the ways structures are represented and in the method used to reduce the search space for efficiency.

The comparison of relationships among features in one structure relative to another is a powerful addition to any structural comparison procedure (see ref. 7 for an excellent discussion). Relationships, such as patterns of hydrogen bonding, involve the comparison of a minimum of two residue positions for every hydrogen bond in both structures. In certain cases, e.g. in the method of Taylor and Orengo (10), relationships—in this case inter-atomic vectors, are compared using their novel double dynamic programming method. More often, the matching of relationships is treated as a combinatorially-intensive task. There are lots of candidate pairs of hydrogen bonds in each structure and matching them relies on methods such as simulated annealing, Monte Carlo simulations and genetic algorithms.

Protocol 3

The alignment: determination of equivalent pairs

Methods used to find the optimal match between entire structures or between parts of structures consisting of the best matching regions of the structures. Equivalent or matched positions are defined by the user (i.e. property distances within a cut-off value) or by the strategy of the method employed (e.g. all matched positions produced by dynamic programming methods).

Dynamic programming methods (6)

1 Construct a matrix with dimensions equivalent to the lengths of the structures to be compared.

2 Each cell in the matrix corresponds to a residue in the first protein matching a residue in the second protein. The matrix accommodates all possible alignments.

3 Cells are filled in with a score relating each two matched positions. These scores may be distances between Cα-atoms, for example, distance scores based on other

Protocol 3 continued

features of the protein (see Table 1), or similarity scores derived from distances. In this description, we will refer to a matrix filled with similarity scores derived from distances.

4 Beginning at one corner (amino-terminal end or carboxy-terminal end of the sequences) of the matrix and heading towards the opposite corner, sum diagonal values to the current position if they are the best score (a residue–residue match), or sum with an off-diagonal score minus a penalty (indicates a possible gap in one protein or the other).

5 The largest value found at one edge of the matrix specifies the first two aligned positions and gives the optimal alignment score for the comparison.

6 The full alignment that produced the optimal score can be traced beginning at the highest value and progressing towards the opposite side of the matrix by following the next best score in the matrix. When the next highest value is on the diagonal, residues are matched in sequence; when an off diagonal score (less a penalty) is the next best choice, then a gap is indicated.

7 This method produces the full alignment including gap regions, but elements within a cut-off value can be used to determine the rigid-body superposition of the structures.

Clique detection methods (23, 25)

1 Represent each structure as a graph of nodes (Cα-atoms or secondary structure elements) and vertices connecting the nodes. Each vertex is a distance between the connected two nodes (atoms).

2 List for each vertex in structure A all such vertices in structure B, which are similar within an error threshold (i.e. vertices connecting the same kind of nodes with similar distances).

3 Find the maximal common sub-graph (largest set of nodes and vertices, which exists in both structure graphs) using a tree search algorithm, Monte Carlo simulation, or a genetic algorithm. Each vertex in the common sub-graph corresponds uniquely to one vertex in both structures A and B.

4 The nodes included in the sub-graph are equivalent for the two structures. If the nodes are atoms, the superposition can be made directly (see Protocol 2). Also, the secondary structure elements can be superimposed as if they were atoms of a rigid molecule, or the Cα-atoms within the SSEs can be superimposed.

Match list approaches (22)

This method is a variation of the clique detection method, which assumes that the structures are initially superimposed, but equivalent matches are not known.

1 In the case of Cα-Cα distance comparisons, create two lists, one for each protein A and B.

(a) In one list, tabulate all Cα-atoms in protein B with matches within a cut-off distance, say 3.5 Å, to a position in protein A.

Protocol 3 continued

(b) In a second list, tabulate all Cα-atoms in protein A with matches within a cut-off distance to a position in protein B.

2 Filter from the list the poorest matches to reduce the number of matches to a unique set of equivalent matches:

(a) Remove matches that are not part of a contiguous run of at least 4 Cα-atoms.

(b) Reduce multiple matches from one protein to a single Cα-atom in the other protein, e.g. does one of the matches extend a contiguous run of existing matches?

(c) If there are still multiple matches remaining, then the match with shortest distance is kept and the others are removed.

Comparisons of relationships (7, 10, 11, 21)

1 The matching of relationships among features of one structure with relationships among features of another structure is accomplished using one of several different techniques.

(a) Monte Carlo simulations (11, 21).

(b) Simulated annealing (7, 21).

(c) Double dynamic programming (10)

(d) Genetic algorithms (22, 26, 27) can also be used.

2 The matched relationships may be insufficient in themselves to accurately align the 3-D structures, and thus would be combined with the feature comparisons within a dynamic programming procedure, for example, to give the final alignment (7).

2.5 Quality and extent of structural matches

Once a structural alignment has been made, a score or scores can be assigned to the alignment that give an indication of the quality and the extent of matching between the two structures. With methods that iteratively improve a structural comparison, an evaluation score is necessary to monitor the improvement at each cycle of comparison, and to indicate when the program should stop because no further improvement in the alignment could be obtained. The final alignment scores can be used to compare different protein comparisons within a family and provide useful indications of the phyletic ancestry of the proteins (e.g. 8, 28). Among the most frequently used key indicators of the 'goodness' of a structural comparison include the root mean squared deviation (RMSD), the number of topologically-equivalent atoms matched in the comparison, and the alignment score that is obtained.

2.5.1 Root mean squared deviations

The RMSD is commonly used to indicate the goodness of fit between two sets of co-ordinates. Often, but not always, the RMSD value is quoted for only those matched Cα-atoms that are within a specified distance cut-off, say atoms within 3.0 Å of each other after the proteins have been superposed. In this case and given the cut-off value of 3 Å, the RMSD obtained and each of the Cα–Cα atom distances contributing to the RMSD will be less than the 3 Å. Alternatively, the RMSD can be calculated over all matched Cα–atom pairs, regardless of the distance between the superposed atoms. Of course, the RMSD can also be calculated between sets of any type of superposed atoms, not just Cα–atom pairs as illustrated in *Protocol 4*.

Protocol 4

Root mean squared deviations (RMSD)

The RMSD gives a measure of the average level of deviations over the matched atoms that are included in the calculation. Given the same number of equivalent atom pairs, a smaller value indicates a better superposition than does a larger value.

Data required

- Co-ordinates of the equivalent sets $\overset{\omega}{\ddot{A}}{}_i^{eq}$ (*trans.*), $\overset{\omega}{\ddot{B}}{}_i^{eq}$ (*trans.,rot.*).

Method

1 Calculate the Euclidean distance between each pair of equivalent atoms $\overset{\omega}{\ddot{A}}{}_i^{eq}$ (*trans.*) and $\overset{\omega}{\ddot{B}}{}_i^{eq}$ (*trans.,rot.*).

2 Take the sum of all squared distances *D*, and divide by the number of pairs, *N*, to give the mean.

3 Calculate the square root of the mean squared distance to obtain the RMSD.

4 Thus, the

$$\mathrm{RMSD} = \sqrt{\sum_{i=1}^{N} D_i^2/N}$$

2.5.2 Topological equivalent atoms pairs

Another criterion that is used to gauge the extent or quality of a superposition of two structures is the number of atom pairs that superpose within a distance cut-off. Structure comparison methods usually try to maximize the number of superposed equivalent atoms while minimizing the RMSD over those equivalent atoms.

Note that two different sets of superposed structures, given the same cut-off value, can have the same number of equivalent matches, but with different RMSD values over those matches. The match with the lower RMSD would be considered the more similar pair. Conversely, one structural comparison, for example, may produce 121 matches with an RMSD of 2.1 Å, while a second comparison matches 50 atom pairs with an RMSD of 1.2 Å: the comparison with the 121 matches would be considered the better match.

2.5.3 Structural alignment scores

For structural alignment methods that employ dynamic programming in order to produce a complete alignment of the structures, including gaps, a key measure of the alignment quality is the alignment score corresponding to the overall optimal structural superposition. This value includes scores for matching all positions and penalties for every gap that appears in the alignment. The alignment score is composed of the values placed into the matching matrix during the dynamic programming procedure. In the case of Cα-atom based comparisons, the residue-residue matching scores would be the distances between the atoms. In the case of procedures that consider other criteria, e.g. those features listed in *Table 1*, the alignment score would include the scores attributed to matches of residue positions according to those features.

The raw alignment score is useful during iterative procedures to provide an indication of the progress of the superposition. Within a family of homologous 3-D structures, the alignment score, normalized for the length of the smaller protein or for the number of matched residues along the sequences, can be compared to give an idea of the mutual structural relationships among the family members.

3 The comparison of identical proteins

3.1 Why compare identical proteins?

The simplest type of comparison of 3-D structures involves the comparison of two (or more) sets of co-ordinates for the same protein. Self-comparisons are often used to reveal:

(a) Similarities/differences between independent solutions of crystal structures.

(b) Similarities/differences among sets of structures, generated using distance geometry, and consistent with distance information obtained in NMR spectroscopy.

(c) Similarities/differences between structures obtained using X-ray diffraction and NMR spectroscopy.

(d) Similarities/differences that occur between apo- and holo-protein structures: alterations in structure that occur upon binding ligands, cofactors, metal ions, etc.

(e) Similarities/differences of two structures after superposing on an identical ligand or subset of residues or co-ordinate positions.

3.2 Comparisons

In the comparison of identical proteins that have 3-D structures that differ to varying degrees, it is needed to compare the structures using a rigid-body approach one time only (*Protocol 5*). No iteration is necessarily required to achieve the best result, since one would typically supply all atom positions in the structure for comparison. Likewise, no pre-comparison is necessary to supply a seed set of residues for the comparison. In practice, iterative procedures are used. Again, if big differences in the structures are anticipated, e.g. the relative domain movements in *Figure 2*, then this approach may not be appropriate except to provide an RMSD value that is an indication of the relative changes to the structures.

Protocol 5

Similarities among different structures of identical proteins

Finds regions of high structural similarity between different solutions of the structure of the same protein.

Data required

- Co-ordinates, minimum of Cα-atoms, for all structures

Method

1 No alignment between the proteins is necessary[a] since the proteins are identical and each position maps 1:1 in sequence along the protein.

2 A single application of a comparison algorithm (see Protocol 2) is sufficient to obtain the optimal result over all of the compared atoms.

3 Calculate the RMSD over all atoms or those within the cut-off distance, as desired (see Protocol 4).

4 Iterative methods (see *below*), seeded by some key positions, can be used also.

5 By adjusting the cut-off value used to define equivalent matched atoms to lower values, the most similar structural regions may be identified and hence, the differences pinpointed too.

[a] Note that different data sets from different sources do not necessarily contain the same amino acids or atoms for the same protein.

4 The comparison of homologous structures: example methods

4.1 Background

Most comparison programs are designed to compare non-identical homologous structures, but they can be also used to superpose structures for the same proteins as described in Section 3. There are a large number of different programs and strategies that have been published and we have necessarily had to select just a few as illustrations—our apologies to any author who feels that we have neglected their own work. In general, the methods fall into two different groups:

(a) Those that require the advance definition of pairs of suspected 'equivalent' atoms in order to seed the alignment. An iterative procedure is then used to maximize the number of equivalent matched atom pairs while minimizing the RMSD.

(b) Those methods that sample the realm of possible solutions and, as a result, automatically find optimal alignments without specifying an initial starting alignment.

Some of these procedures involve rigid-body comparisons and others generate a full alignment of the sequences on the basis of the structures. In *Figure 3*, we show the extreme differences in results for the same proteins obtained with a multiple sequence alignment, a rigid-body structure comparison, and a procedure (7) that combines the comparison of properties and relationships to derive the structural matching.

(a) F-STR

```
               *****          ****           ***********  ******
4APE-N    ---STCGSATTTPIDSLDDAYITPVQ-IGT-----PAQTLNLDFDTGSSDLWVFSSETTASEVDGQTIYTPSK
2APP-N    --AASGVATNTPTA-NDEEYITPVT-IG-------GTTLNLNFDTGSADLWVFSTELPASQQSGHSVYNPSA
2APR-N    --AGVGTVPMTDYG-NDIEYYGQVT-IGT-----PGKKFNLDFDTGSSDLWIASTLCT-NCGSGQTKYDPNQ

4APE-C    YTGSITYTAVSTKQ---GFWEWTSTGYAVGSGTFKSTSIDGIADTGTTLLYLPATVVSA---------YWAQ
2APP-C    YTGSLTYTGVDNSQ---GFWSFNVDSYTAGSQ-SGDG-FSGIADTGTTLLLLDDSVVSQ---------YYSQ
2APR-C    FKGSLTTVPIDNSR---GWWGITVDRATVGTSTVAS-SFDGILDTGTTLLILPNNIAAS---------VARA

                                              ****  *
4APE-N    STTAKLLSGATWSISYGDGSSSSGD----VYTDTVSVGGLTVTGQ-------------------AVESAKKVS
2APP-N    --TGKKLSGYTWSISYGDGSSASGN----VFTDSVTVGGVTAHGQ-------------------AVQAAQQIS
2APR-N    SSTYQAD-GRTWSISYGDGSSASGI----LAKDNVNLGGLLIKGQ-------------------TIELAKREA

4APE-C    VSGAKSSSSV---------GGYVFPCSA-TLPSFTFGVGSARIVIPGDYIDFGPISTGSSSCFGGIQSSA---
2APP-C    VSGAQQDSNA---------GGYVFDCST-NLPDFSVSISGYTATVPGSLINYGPSGD-GSTCLGGIQSNS---
2APR-C    Y-GASDNGD----------GTYTISCDTSAFKPLVFSINGASFQVSPDSLVFEEF---QGQCIAGFGYG----

                    ****                              ****           ****
4APE-N    SSFTEDSTIDGLLGLAFSTLNTVSPTQQKTFFDNAKAS--LDSPVFTADLGY---HAPGTYNFGFIDTTA
2APP-N    AQFQQDTNNDGLLGLAFSSINTVQPQSQTTFFDTVKSS--LAQPLFAVALKH---QQPGVYDFGFIDSSK
2APR-N    ASFASG-PWDGLLGLGFDTITTVRG--VKTPMDNLISQGLISRPIFGVYLGKAKNGGGGEYIFGGYDSTK

4APE-C    ------GIGINIPGD---------------VALKAA----------FVVFNGA-----TTPTLGFASK----
2APP-C    ------GIGFSIPGD---------------IFLKSQ----------YVVFDSD-----G-PQLGFAPQA---
2APR-C    ------NWGFAIIGD---------------TFLKNN----------YVVFNQG-----V-PEVQIAPVA--E
```

(b) SEQ

```
4APE-N    -STGSATTTPIDSLD--------DAYITPVQIGT-P-AQTLNLDFDTGSSDL-----------WVFSSETTAS
2APP-N    AASGVATNTPTAN-D--------EEYITPVTIG----GTTLNLNFDTGSADL-----------WVFSTELPAS
2APR-N    AGVGTVPMTDYGN-D--------IEYYGQVTIGT-P-GKKFNLDFDTGSSDL-----------WI-ASTLCTN

4APE-C    -YTGSITYTAVSTKQGFWEWTSTGY--AVGSGTFK-STSIDGIADTGTTLLYLPATVVSAYWAQVSGAKSS
2APP-C    -YTGSLTYTGVDNSQGFWSFNVDSYTAGSQSG-----DGFSGIADTGTTLLLLDDSVVSQYYSQVSGAQQD
2APR-C    -FKGSLTTVPIDNSRGWW----GITVDRATVGTSTVASSFDGILDTGTTLLILPNNIAASV-ARAYGASDN


4APE-N    EVDGQTIYT-PSKSTTAKLLSGATWSISYG------DGSS---SSGDVYTD--TVSVGGLTVTGQAVESAKK
2APP-N    QQSGHSVYN-P--SATGKKLSGYTWSISYG------DGSS---ASGNVFTD--SVTVGGVTAHGQAVQAAQQ
2APR-N    CGSGQTKYD-PNQSSTYQA DGRTWSISYG------DGSS---ASGILAKD--NVNLGGLLIKGQTIELAKR

4APE-C    SSVGG--YVFPC-SAT-LP------SFTFG-----VGSARIVIPGD-YIDFGPISTGSSSCFGGIQSSAGI
2APP-C    SNAGG--YVFDC-S-T-N-LPDFSVSIS-GYTATVPGSL--INYGP-SGD-------G-STCLGGIQSNSGI
2APR-C    GD-GT--YTI---SCDTSAFKPLVFSI---------NGASFQVSPDSLVFEEPQ---G-QCIAG----F-GY


4APE-N    VSSSPTEDSTIDGLLGLAFSTLNTVSPTQQKTFFDNAKASLDSPVFTADL---GYHAPGTYNFGFIDTTA
2APP-N    ISAQPQQDTNNDGLLGLAFSSINTVQPQSQTTFFDTVKSSLAQPLFAVAL---KHQQPGVYDFGFIDSSK
2APR-N    EAASFASGPN-DGLLGLGFDTITTVRGVKTPMDNLISQGLISRPIFGVYLGKAKNGGGGEYIFGGYDSTK

4APE-C    GINIPG-----DVALKAAF----VVFNGA-------------TTP----TL----------G---FASK--
2APP-C    GFSIPG-----DIPLKSQY----VVFD-S-------------DGP----QL----------G---FAPQA-
2APR-C    GNWGFAIIG--DTFLKNNY----VVFN-Q-------------GVP-----------------EVQIAPVAE
```

Figure 3 The differences in alignments of the aspartic proteinase amino- and carboxyl-terminal domains (labelled with an 'N' or 'C', respectively) from (a) multifeature (7) and from (b) multi-sequence comparisons. Asterisks in (a) indicate those positions among the structures that were found to be equivalent under rigid-body superposition with the computer program MNYFIT (16). PDB codes: 4APE, endothiapepsin; 2APP, penicillopepsin; 2APR, rhizopuspepsin. (From ref. 8, with permission.)

4.2 Methods that require the assignment of seed residues

As we have already discussed above, a set of seed matches between a pair of structures is often needed by methods in order to initiate the comparison of structures, because some residue properties, such as Cα–Cα distances, require a partially correct alignment in order to calculate these distances. Once seeded, the alignment improves over several rounds of comparison. Obvious candidates for seed residues are listed in *Protocol 6*.

Protocol 6

Finding initial seed residues

Required data

- Sequences and/or co-ordinates of the proteins to be compared

Method

1. Supply a minimum of three conserved residues from a sequence-based alignment, or
2. Supply key residues implicated in a conserved binding or catalytic motif, or
3. Supply segments corresponding to secondary structure elements observed on a graphics device to be conserved between the structures.

In *Protocol 7*, we present a general procedure for the alignment of two structures using rigid-body comparisons, which requires a seed set of matches between the two 3-D structures.

Protocol 7

Semi-automatic methods

Required data

- Co-ordinates of the proteins to be compared
- Initial set of equivalent atom pairs to seed the alignment procedure

Method

1. Calculate translation vector based on seed residues, translate all co-ordinates to the origin and calculate the rotation matrix for the seed residues (see Protocol 2).
2. Apply the rotation matrix to all atoms of the second protein to achieve the first superposition (see Protocol 2).
3. Obtain the alignment using dynamic programming or clique analysis (see Protocol 3).

Protocol 7 continued

4 For all matched residue pairs in the alignment, calculate the Euclidean distance. Those matched pairs within the distance cut-off value will form the new updated set of equivalent atom pairs for the next round of super-positioning.

5 Repeat steps 1–4 until convergence is obtained: the calculated RMSD (Protocol 4) does not decrease and the number of equivalent atom pairs matched in the two proteins does not increase.

6 A rigid-body comparison has been used as a starting point for more detailed structural comparisons involving multiple structural features (e.g. the program COMPARER described in ref. 7).

4.3 Automatic comparison of 3-D structures

To get around the requirement for an initial set of equivalent seed matches, alternative methods have been developed. Here, several of the many published methods are described to illustrate the different strategies that have been employed:

(a) Methods that supply seed matches automatically to a rigid-body approach after making a sequence-based alignment (*Protocol 8*).

(b) Methods that use a genetic algorithm (*Protocol 9*) or 'spectra'-comparison method (*Protocol 10*) to find the optimal rigid-body comparison.

Protocol 8

Structural comparisons seeded from sequence alignments

Automatic alignment of two homologous protein structures without the need to specify initial equivalent atoms pairs. Method can fail for proteins of low sequence similarity.

Required data

- Cα-atom co-ordinates of the compared structures and their sequences

Method

1 Align the amino acid sequences with a dynamic programming algorithm (Protocol 3, but using sequence-matching scores to produce the alignment).

2 Superimpose the structures according to Protocol 7 using the most conserved portions of the sequence alignment as the initial set of seed residues.

(c) Methods that do not make rigid-body comparisons directly, but instead make comparisons on the basis of similarities in structural properties and/or relationships (*Protocols 11–14*).

4.3.1 Structural comparisons seeded from sequence alignments

Russell and Barton (9) developed a method that first makes an alignment of the sequences and then uses equivalent matches defined in the alignment to seed a structural comparison. We have given a general protocol for such an approach above (*Protocol 8*). The procedure should work well for proteins where portions of the sequence alignment can be trusted; when the sequence similarity is low and the alignment is not correct, then the method may not be useful.

4.3.2 Rigid-body comparisons using a genetic algorithm

Genetic algorithms (29) describe the solution to a problem within a numerical string. A large number of strings are originally assigned random values as their solutions, and the genetic algorithm seeks to evolve this initial set towards better and better solutions by exchanging partially good solutions among strings and by mutating the strings. *Protocol 9* describes the general procedures used by May and Johnson (26, 27) to automatically compare one or more structures. The approach is time-consuming but has been successfully adapted to parallel processors (Lehtonen and Johnson, unpublished).

Protocol 9

GA_FIT (ref. 26, 27)

Automatic rigid-body alignment of two protein structures without the need to specify initial equivalent atoms pairs.

Required data

- Cα-atom co-ordinates of compared structures

Method

1 Create a large random set of superpositions for the pair of structures.

2 Assign equivalent matches (Cα-atoms within a specified distance cut-off) using dynamic programming and score each alignment (see Protocol 3).

3 Create a new set of superpositions by crossing-over and mutating the existing solutions.

4 Repeat steps 2–3 until a close to final solution is achieved.

5 Optimize the best found superposition/alignment by least squares minimization (see Protocol 2).

6 Calculate the final alignment with the dynamic programming algorithm (see Protocol 3).

4.3.3 Rigid-body comparisons using the density of Cα-atom packing and spectral alignment

VERTAA, described in *Protocol 10* (Lehtonen and Johnson, unpublished method), compares and aligns spectra equal to the Cα-atom density in each structure as a function of the position along the sequence of each protein (*Figure 4*). This method is a rapid and automatic means for comparing structures.

Protocol 10

Local similarity search by VERTAA

Fast, automatic alignment of two protein structures without the need to specify initial equivalent atoms pairs.

Required data

- Cα-atom co-ordinates of the two structures.

Method

1 VERTAA, for each of two structures, plots the number of Cα-atoms within a given radius (14.0 Å) from each Cα-atom in the structure. Other properties can be used too.

2 These 'spectra' are scaled and overlapping segments are aligned. More than one alignment method is available:

(a) The dynamic programming algorithm (Protocol 3). Fast and robust if the input values are properly scaled.

(b) The Fourier correlation (21). The values can be considered as a function over a limited range and a correlation function obtained with the fast Fourier transform to bring the spectra into register. Dynamic programming is then used to define equivalent matches (Protocol 3).

3 Superimpose the structures (see Protocol 2) based on equivalent matches defined in step 2.

4 Define a new alignment with dynamic programming and the Cα-Cα distances of the superimposed structures within 3.5 Å (see Protocol 3).

5 While the alignment and superimposition improve, repeat steps 3 and 4.

4.3.4 Structural comparisons based on matching Cα-atom contact maps

Holm and Sander (11) make comparisons by comparing Cα atom–Cα atom contact maps (by contacts, we mean nearby in space) constructed from each protein structure (*Protocol 11*).

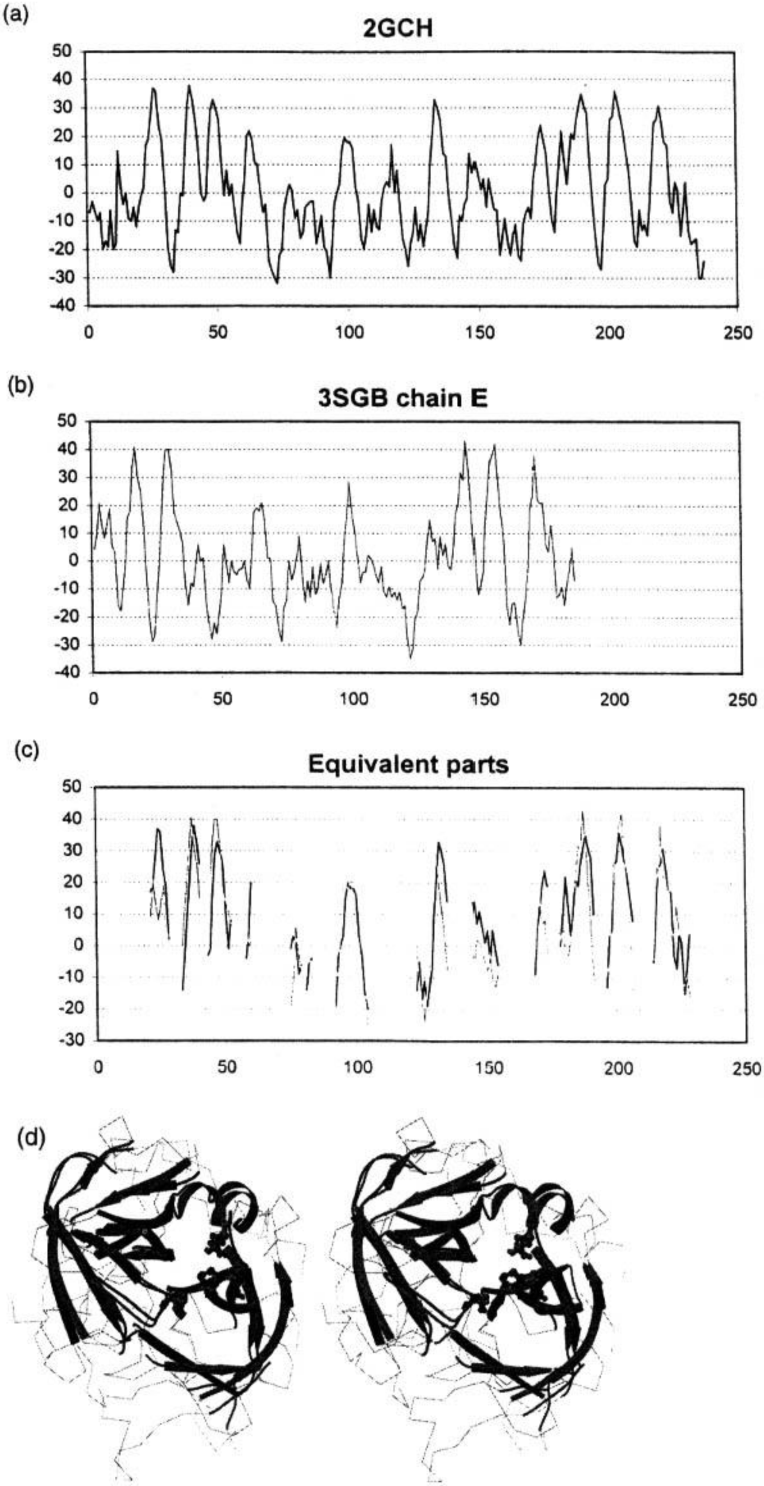

(a)
2GCH
50
40
30
20
10
0
-10
-20
-30
-40
0
50
100
150
200
250
(b)
3SGB chain E
50
40
30
20
10
0
-10
-20
-30
-40
0
50
100
150
200
250
(c)
Equivalent parts
50
40
30
20
10
0
-10
-20
-30
0
50
100
150
200
250
(d)

Figure 4 Plots of Cα-atom densities, alignment of plots, and the corresponding superposition of the structures. (a) Cα-atom densities of residues in γ-chymotrypsin A (PDB code 2GCH). (b) Cα-atom densities of residues in *Streptomyces griseus* proteinase B (PDB code 3SGB, chain E). For both spectra, the average density is set equal to 0. (c) The parts of the two plots from (a) (dark) and (b) (light), which correspond to each other according to the alignment of their spectra. (d) The superpositioned 3-D structures (2GCH dark, 3SGB light) based on the alignment specified in (c). The side chains of the catalytic triad are shown and the closely matching parts are drawn as ribbon diagrams. This superposition was made with the computer program VERTAA (Lehtonen and Johnson, unpublished results) and contains 118 residues within 3.5 Å with an RMSD of 1.8 Å. Figure (d) was prepared with MOLSCRIPT (52).

Protocol 11

Structure comparison by DALI (11)

Automatic alignment by finding the optimal clique for contact maps obtained from the structures (Protocol 3).

Required data

- Cα-atom co-ordinates of compared structures

Method

1 Calculate a distance matrix for each protein A. Element (i, j) of the matrix contains the intramolecular distance between the i^{th} and j^{th} Cα-atom in A. Likewise, calculate a distance matrix for protein B.

2 List from each distance matrix all possible 6 by 6 sub-matrices.

3 Reduce the number of sub-matrices by clustering similar ones and using the mean of each cluster as the contact pattern. Sort contact patterns by intra-pattern distance.

4 Compare each pair of two contact patterns from A with all pairs of sub-matrices from B. Compare each pair of two contact patterns from B with all pairs of sub-matrices from A. List all pair–pair matches.

5 Remove redundancy from the list of matches and sort it by match quality, which is a function of the differences between the sub-matrices from A and from B.

6 Find the most extensive, non-exclusive collection of matches from the list. DALI uses a Monte Carlo simulation to search the best 40 000 matches. The simulation tries to extend the matches by combining matches that contain a common contact pattern in both distance matrices. The random element of the simulation is used to find the best scoring combination from mutually exclusive possibilities.

4.3.5 Comparisons using double dynamic programming

Taylor and Orengo (10) have developed a novel use of dynamic programming in order to facilitate the comparison of relationships. Dynamic programming is used once to compare structural relationships in the two proteins thus providing scores for a second round of dynamic programming where the two structures are aligned (*Protocol 12*).

Protocol 12

Structure comparison by SSAP (10)

Fast automatic alignment of two protein structures using double dynamic programming.

Required data

- Cα-atom co-ordinates of compared structures

Method

1 Calculate a distance matrix for protein A. Element (*i*, *j*) of the matrix contains the intramolecular vector from the i^{th} to the j^{th} Cα-atom in A. The vector is in the co-ordinate frame defined by the covalent bonds of A's i^{th} Cα-atom. Likewise, calculate a distance matrix for protein B.

2 Calculate intramolecular difference matrices for each pair of rows from the two distance matrices. Thus, element (*i*, *j*) of the matrix constructed from row *h* of A's, and row *k* of B's distance matrix will contain the difference of the magnitude of the vectors $\overset{\omega}{A}_{hi}$ and $\overset{\omega}{B}_{kl}$ converted to a similarity value.

3 Low level alignments of the local structure are made first using a dynamic programming algorithm (see Protocol 3) to find the best scoring path through the intramolecular difference matrices (see ref. 10 for details). Scores along the path will contribute to a separate 'summed scoring matrix' from which the final alignment will be determined.

4 Use a dynamic programming algorithm (see Protocol 3) to trace an alignment path through the summed scoring matrix. This higher level alignment defines the equivalent matches between the structures.

4.3.6 Structural alignments based on secondary structure element (SSE) matching

Kleywegt and Jones (30) describe a method for structural comparisons based on the alignment of elements of regular secondary structure (*Protocol 13*).

Protocol 13

Structure comparison by DEJAVU (30)

Automatic alignment of protein structures by finding the optimal clique on the basis of secondary structure comparisons (Protocol 3).

Required data

- Cα-atom co-ordinates of the two structures or SSE templates of the structures

Generation of SSEs with YASSPA in O (see ref. 30 for details)

Search the structures and tabulate main chain fragments that are similar to templates of typical α-helices and β-strands.

Protocol 13 continued

Comparison of structures with DEJAVU (see ref. 30 for details)

1 Check that both structures have the required number of SSEs.

2 Check that there exists at least one SSE of the same type (same length—number of Cα-atoms) in the second structure for each SSE in first structure.

3 Find the most extensive, non-exclusive collections of matched SSEs. DEJAVU does a depth-first tree search to find all sets of matching SSE pairs, where all pairs in a set are matching also in 3-D space. The tree contains all possible combinations of pairs. If the path from the root to a node already has too many mismatches, the sub-tree below the node is not searched, saving time.

4 Report the matched SSEs and the Cα-atoms for the best scoring alignment.

5 The output can be directed to external programs for refinement of the superposition and visualization.

4.4 Multiple structural comparisons

Multiple structural comparisons can be made using several different strategies. Sutcliffe *et al.* (16) constructed multiple rigid-body structural alignments by comparing each structure to an average representation of the structures (in practice, one of the structures was chosen for this purpose at the beginning of the comparisons). More frequently, multiple alignments are assembled from pairwise structural alignments according to the topology of a tree estimated on the basis of sequence alignments (9, 27, *inter alia*). This (*Protocol 14*) follows the strategy first introduced by Barton and Sternberg (31) and Feng and Doolittle (32) for the efficient multiple alignment of protein sequences.

Protocol 14

Multiple structural alignments from pairwise comparisons

Multiple alignments assembled from pairwise comparisons.

Required data

- Cα-atom co-ordinates of the structures in PDB format

A general approach

1 Use a sequence alignment procedure to align the proteins and to cluster them as a bifurcating tree (see refs 31–34 and several chapters in ref. 35).

2 Use a pairwise structural alignment method to align clusters according to the tree topology. This will involve comparing pairs of structures, one structure with a set of previously aligned structures, and aligned structures with aligned structures, until all clusters have been coalesced into a final alignment involving all of the proteins.

5 The comparison of unrelated structures

5.1 Background

Non-homologous protein structures have frequently been compared to highlight features of protein structure that are common across many families. It has been less often recognized, however, that proteins with different folds can also share similarities that can extend to a fairly large organization of their structures, for example about common ligand and cofactor binding sites. The elements contributing to these similarities are likely to involve fragments of each structure that do not map along the protein chains in any predetermined way (*Figure 5* and ref. 22). Thus, despite similar local structure, the segments contributing to the similarity can be both rearranged and discontinuous with respect to each other (e.g. 36–38). Such similarities are particularly difficult to recognize even if a hint of a common functional requirement is present, like a common cofactor. Nonetheless, the recognition of local similarities can provide evidence about the rules governing the structure-function relationship suitable for protein modelling, the prediction of structure from sequence and computer-based drug design. For example, Kobayashi and Go (39) have reported a local motif about the ATP binding site common to cyclic-AMP dependent protein kinase and D-Ala:D-Ala ligase involving 4 equivalent residues. Comparisons using the computer program GENFIT (22) automatically and repeatedly found up to 60 matches (36) that includes an extensive supersecondary structure organization used to position polar and nonpolar residues that interact with the similarly oriented cofactor, bound metal and bound water molecules (*Figure 6*).

Given two unrelated protein structures, A and B, the goal of a computer program is to find the largest equivalent subset of the two structures. Because the proteins are not derived from a common ancestor, the matches providing equivalent structural interactions:

(a) Are not necessarily sequential along the two sequences (*Figure 5*).

(b) Can involve matched elements of secondary structure whose chain directions are opposite to each other (*Figure 5*) but can still provide equivalent interactions, for example, with bound ligand.

Here we describe two different approaches that have been successfully used to find similarities among unrelated protein structures, SARF2 (40, 41) and GENFIT (22). SARF2 (41) considers SSEs, finds the maximal common sub-graph for two structures; and systematically creates different alignments, tries to improve them, evaluates them and reports the best found alignments (*Protocol 15*). GENFIT (22) considers matched segments of Cα-atoms and employs a genetic algorithm to randomly sample large numbers of possible alignments and uses a match-list approach to assign equivalent segments of structure, which are subsequently used to make a local rigid-body superposition for each alignment (*Protocol 16*). GENFIT, by virtue of the genetic algorithm, will find and report different equally likely superpositions in different runs (*Figure 7*).

Both of these approaches establish equivalent matches between objects,

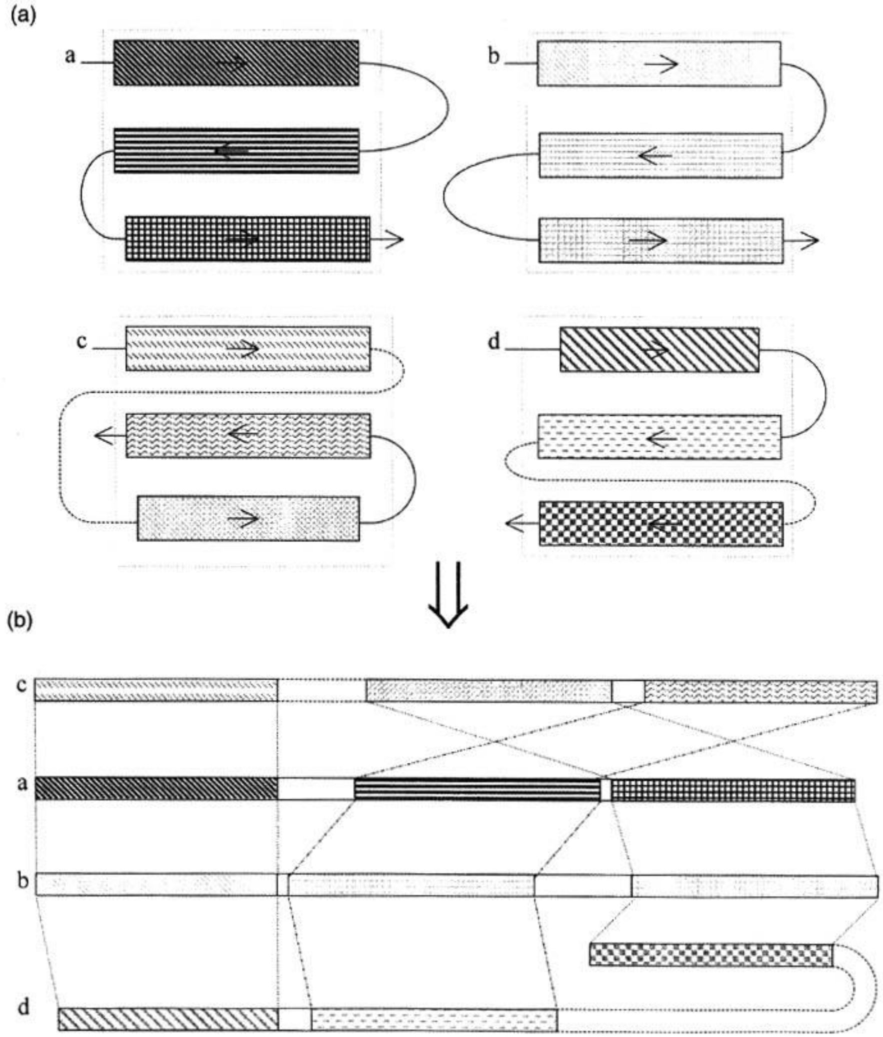


Figure 5 Illustration of the matching of local structural similarities from non-homologous proteins *versus* homologous proteins. (a) Topological diagrams show four protein structures (*a*, *b*, *c*, and *d*) with similar local structural elements. Topologies *a* and *b* represent two homologous structures with the same fold, while *c* has a different topology than *a* and *b*, yet has the same core structure. Topology *d* illustrates a different fold that still has the structurally equivalent segments of polypeptide chain in same place, but some segments may have opposite chain directions. In (b), the correspondence found in the structural alignment is shown at the sequence level. Note that only the sequences of *a* and *b* have a straightforward linear correspondence. (From ref. 22, with permission.)

segments of Cα-atoms (GENFIT) or SSEs (SARF2). GENFIT starts with 'too many' equivalent matches and reduces them until a maximal, but non-conflicting set, is obtained. This is done for each of the many parallel comparisons being made, but a single optimal result will be obtained in any one run: the parallel com-

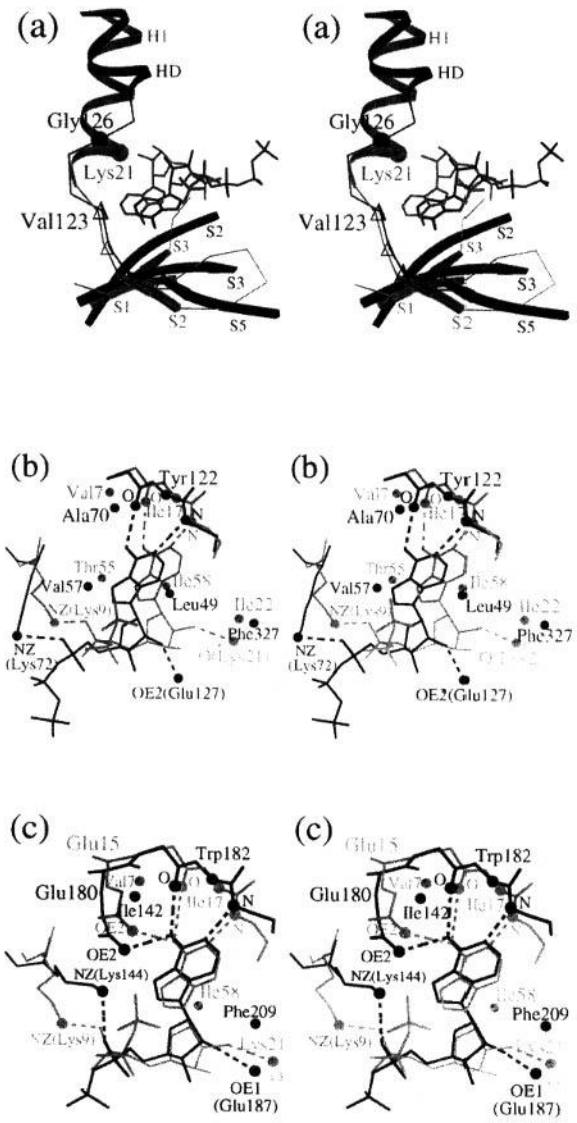


Figure 6 Local similarity of ATP cofactor binding site seen in the pairwise superposition of ribonucleotide reductase (PDB code: 3R1R, chain A; light grey) with cAMP-dependent protein kinase (1CDK; dark grey) (a and b), and with D-Ala:D-Ala ligase (1IOW; dark grey) (c). The bound ATP molecules of the structures are shown as stick models. In (a), the four common segments are drawn as ribbon diagrams. (b and c) The environment around the cofactor is illustrated by showing the equivalent hydrogen bonds (dashed lines) and equivalent C^{α}-atoms (spheres) forming hydrophobic contacts to the cofactor. (From ref. 37, with permission.)

parisons converge towards that result. SARF2 searches among a large set of matches between the structures and finds the largest non-conflicting subset of matches. Both methods are free from restraints on the order and chain direction of objects along the sequence, but optional restraints can be applied.

Protocol 15

Structure comparison by SARF2 (41)

Local similarity alignment of non-homologous structures.

Required data

- Cα-atom co-ordinates of the two structures

Method

1 Search for and tabulate main-chain fragments from the structures that are similar to five-residue long templates of typical α-helices and β-strands.

2 Create a list of SSE pairs from the first structure that match SSE pairs from the second structure. Distance and angular criteria between the SSE's in both structures is important to the determination of a match.

3 Combine matches to find the largest collection of SSE's that can be aligned. SARF2 uses an exhaustive, recursive search algorithm to find possible solutions (see ref. 41 for details).

4 For the best solutions found, superimpose the matched SSE's and then add nearby Cα-atoms to matched regions using the dynamic programming method. Iteratively repeat the superpositions of Cα-atoms until the maximum number of matched atoms have been found.

5 A list of superpositions, ranked according to an alignment score, result.

Protocol 16

GENFIT (22)

Automatic alignment of two locally similar protein structures using a genetic algorithm. This implementation has been designed for parallel processing environments.

Required data

- Cα-atom co-ordinates of the two structures

Method

1 Create a large random set of superpositions for the pair of structures.

2 Assign equivalent matches using the match list algorithm (see Protocol 3). Criteria for a match include:

(a) Cα-atom matches must be within a user specified distance cut-off.

(b) Matches must include a minimum of four consecutive Cα-atoms.

(c) The direction of the main chain for matched segments is unimportant by default.

(d) Matches do not need to be co-linear (i.e. the location of a match along the sequence relative to other matches is unimportant).

Protocol 16 continued

3 Calculate an alignment score for each superposition and create a new set of superpositions by crossing-over and mutating existing ones (see ref. 22 for details).

4 Repeat steps 2 and 3 until convergence has been achieved.

5 Optimize the best superposition/alignment by least-squares rigid-body minimization (Protocol 2).

6 Recalculate the alignment with the match list algorithm (Protocol 4).

7 If the number of equivalent matches has increased or the fit has improved, repeat steps 5 and 6 with the current alignment.

8 Repetitive runs can produce different results showing that equally likely alternative results exist.

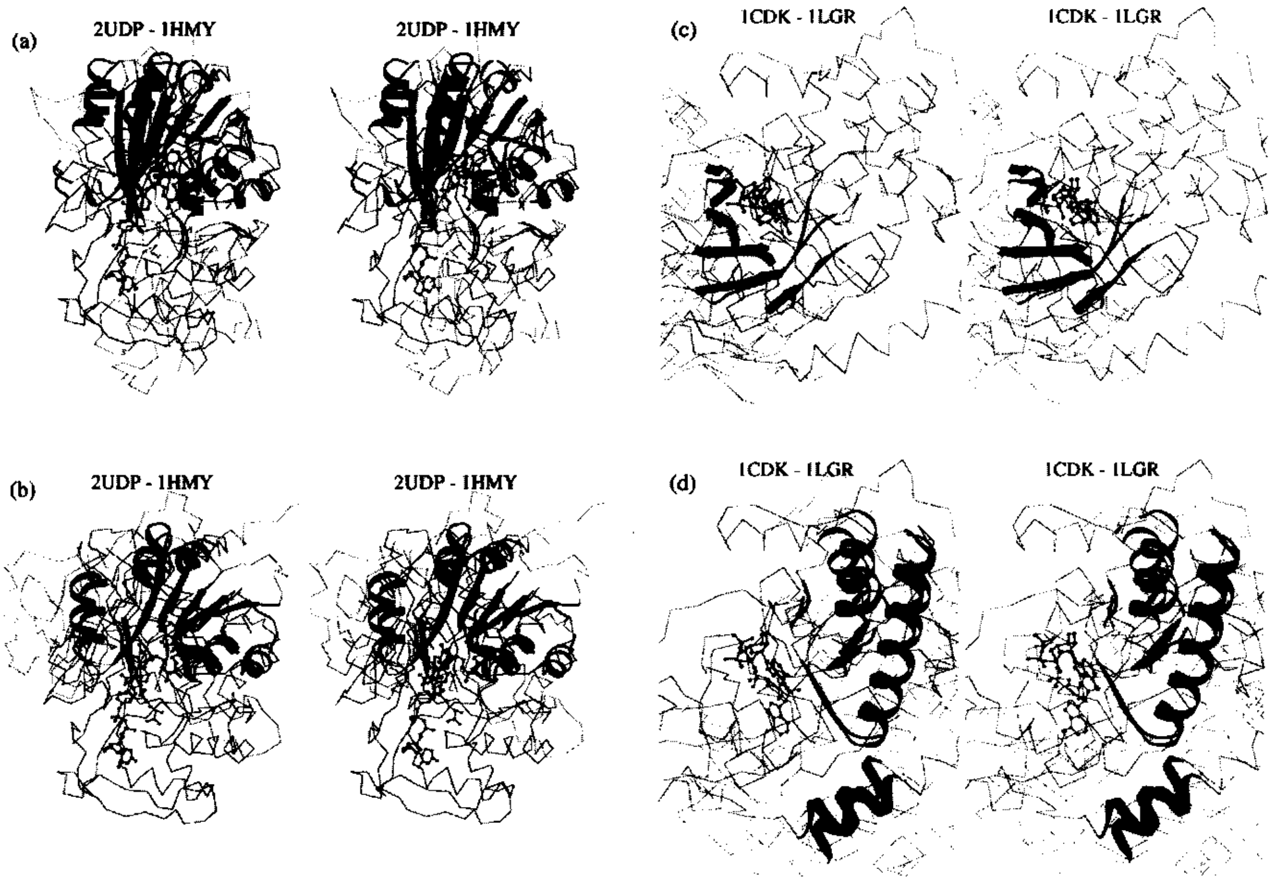
6 Large-scale comparisons of protein structures

One of the straightforward goals in bioinformatics today is to compare, cluster and classify both sequences and the known 3-D structures. Initially, this means categorizing each existing sequence or structure in a data bank. Then, when new entries are made to sequence and structure data banks, each new entry will need to be compared against the existing classifications.

The methods described in this chapter can and have been applied to such analyses. For example, both MNYFIT (16) and COMPARER (7) have been used to accurately align all families of 3-D structures containing two or more structures (43–45), that can be accessed in a public database: *http://www-cryst.bioc.cam.ac.uk/cgi-bin/joy.cgi*. Other available databases include FSSP (46) created using DALI (11): *http://www.ebi.ac.uk/dali/fssp/*; and CATH (47) created in part using SSAP (10): *http://www.biochem.ucl.ac.uk/bsm/cath/*. Several other data banks worth mentioning include MMDB (48): *http://www.ncbi.nlm.nih.gov/Structure/* and SCOP (49): *http://scop.mrc-lmb.cam.ac.uk/scop/*

In *Figure 8*, we present a classification of structures from several different families that belong to the all-β structural classification. This classification was made by comparing the structures on the basis of their secondary structures and then clustering them according to the pairwise structural similarity (44, 50).

Figure 7 Two examples of differing alignments of locally similar structures. (a and b) Superposition of UDP-galactose 4-epimerase chain B (2UDP) and DNA methyltransferase (1HMY) showing similarity between the larger domains. In (b), 1HMY has been rotated by 180 degrees around the axis of the β-sheet in comparison to (a). The symmetry of the nearly planar β-sheet allows for several different, but similarly-scoring alignments. (c and d) Superposition of cyclic-AMP-dependent protein kinase (1CDK) and glutamine synthetase (1LGR) showing local similarities about the ATP-binding sites and the differences seen from matching fewer longer segments (α-helices) or many shorter segments (β-sheets). In (c), the antiparallel β-sheets are aligned and the cofactors overlap, while in (d) the α-helices are matched, but the β-sheets and the cofactor do not superpose well. The superpositions have been made with program GENFIT (22). (From ref. 22, with permission.)


(a)
2UDP - 1HMY
2UDP - 1HMY
(c)
1CDK - 1LGR
1CDK - 1LGR
(b)
2UDP - 1HMY
2UDP - 1HMY
(d)
1CDK - 1LGR
1CDK - 1LGR

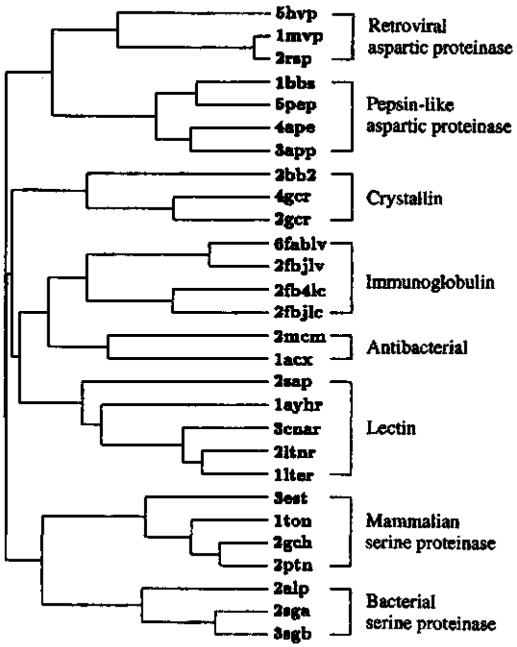


Figure 8 Dendogram of clusters of protein structures composed primarily from β-strands. Each cluster, a family of proteins, is distinguished from the others by its unique fold. (From ref. 44, with permission.)

References

1. Dayhoff, O. M., Barker, W. C., and Hunt, L. T. (1983). In *Methods in enzymology* (ed. C. H. W. Hirs and S. W. Timasheff). Vol. 91, p. 524. Academic Press, London.
2. Chothia, C. (1992). *Nature*, **357**, 543.
3. Blundell, T. L. and Johnson, M. S. (1993). *Protein Sci.*, **2**, 877.
4. Johnson, M. S., Srinivasan, N., Sowdhamini, R., and Blundell, T. L. (1994). *Crit. Rev. Biochem. Mol. Biol.*, **29**, 1.
5. Robertus, J. D., Alden, R. A., Birktoft, J. J., Kraut, J., Powers, J. C., and Wilcox, P. E. (1972). *Biochemistry*, **11**, 2449.
6. Needleman, S. B. and Wunsch, C. D. (1970). *J. Mol. Biol.*, **48**, 443.
7. Sali, A. and Blundell, T. L. (1990). *J. Mol. Biol.*, **212**, 403.
8. Johnson, M. S., Sali, A., and Blundell, T. L. (1990). In *Methods in enzymology* (ed. R. F. Doolittle), Vol. 183, p. 670. Academic Press, San Diego.
9. Russell, R. B. and Barton, G. J. (1992). *Proteins*, **14**, 309.
10. Taylor, W. R. and Orengo, C. A. (1989). *J. Mol. Biol.*, **208**, 1.
11. Holm, L. and Sander, C. (1993). *J. Mol. Biol.*, **233**, 123.
12. McLachlan, A. D. (1972). *Acta Crystallogr.*, **A28**, 656.
13. McLachlan, A. D. (1979). *J. Mol. Biol.*, **128**, 49.
14. McLachlan, A. D. (1982). *Acta Crystallogr.*, **A38**, 871.
15. Diamond, R. (1988). *Acta Crystallogr.*, **A44**, 211.
16. Sutcliffe, M. J., Haneef, I., Carney, D., and Blundell, T. L. (1987). *Protein Eng.*, **1**, 377.
17. Diamond, R. (1992). *Protein Sci.*, **1**, 1279.

18. Shapiro, A., Botha, J. D., Pastor, A., and Lesk, A. M. (1992). *Acta Crystallogr.*, **A48**, 11.
19. Kearsley, S. K. (1989). *Acta Crystallogr.*, **A45**, 208.
20. Kearsley, S. K. (1990). *J. Comput. Chem.*, **11**, 1187.
21. Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (ed.) (1992). *Numerical recipes in C. The art of scientific computing.* (2nd edn). Cambridge University Press, Cambridge.
22. Lehtonen, J. V., Denessiouk, K., May, A. C. W., and Johnson, M. S. (1999). *Proteins*, **34**, 341.
23. Mitchell, E. M., Artymiuk, P. J., Rice, D. W., and Willett, P. (1990). *J. Mol. Biol.*, **212**, 151.
24. Gibrat, J.-F., Madej, T., and Bryant, S. H. (1996). *Curr. Opin. Struct. Biol.*, **6**, 377.
25. Grindley, H. M., Artymiuk, P. J., Rice, D. W., and Willet, P. (1993). *J. Mol. Biol.*, **229 (3)**, 707.
26. May, A. C. W. and Johnson, M. S. (1994). *Protein Eng.*, **7**, 475.
27. May, A. C. W. and Johnson, M. S. (1995). *Protein Eng.*, **8**, 873.
28. Johnson, M. S., Sutcliffe, M. J., and Blundell, T. L. (1990). *J. Mol. Evol.*, **30**, 43.
29. Goldberg, D. E. (ed.) (1989). *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, Reading, MA.
30. Kleywegt, G. J. and Jones, T. A. (1997). In *Methods in enzymology* (ed. C. W. Carter and R. M. Sweet), Vol. 277, p. 525. Academic Press.
31. Barton, G. J. and Sternberg, M. J. (1987). *J. Mol. Biol.*, **198**, 327.
32. Feng, D. F. and Doolittle, R. F. (1987). *J. Mol. Evol.*, **25**, 351.
33. Johnson, M. S. and Overington, J. P. (1993). *J. Mol. Biol.*, **233**, 716.
34. Johnson, M. S., May, A. C. W., Rodionov, M. A., and Overington, J. P. (1996). In *Methods in enzymology* (ed. R. F. Doolittle, Vol. 266, p. 575. Academic Press.
35. Doolittle, R. F. (ed.) (1996). In *Methods in enzymology*, Vol. 266, p. 711. Academic Press, San Diego.
36. Denessiouk, K., Lehtonen, J. V., Korpela, T., and Johnson, M. S. (1998). *Protein Sci.*, **7**, 1136.
37. Denessiouk, K., Lehtonen, J. V., and Johnson, M. S. (1998). *Protein Sci.*, **7**, 1768.
38. Denessiouk, K., Denesyuk, A. I., Lehtonen, J. V., Korpela, T., and Johnson, M. S. (1999). *Proteins*, **35**, 250.
39. Kobayashi, N. and Go, N. (1997). *Nature Struct. Biol.*, **4**, 6.
40. Alexandrov, N. N., Takahashi, T., and Go, T. (1992). *J. Mol. Biol.*, **225**, 5.
41. Alexandrov, N. N. and Fischer, D. (1996). *Proteins*, **25**, 354.
42. Sali, A., Overington, J. P., Johnson, M. S., and Blundell, T. L. (1990). *Trends Biochem. Sci.*, **15**, 235.
43. Overington, J. P., Johnson, M. S., Sali, A., and Blundell, T. L. (1990). *Proc. R. Soc. Lond. B*, **241**, 132.
44. May, A. C. W., Johnson, M. S., Rufino, S. D., Wako, H., Zhu, Z.-Y., Sowdhamini, R., *et al.* (1994). *Phil. Trans. R. Soc. Lond. B*, **344**, 373.
45. Mizuguchi, K., Deane, C. M., Blundell, T. L., Johnson, M. S., and Overington, J. P. (1998). *Bioinformatics*, **14**, 617.
46. Holm, L. and Sander, C. (1998). *Nucleic Acids Res.*, **26**, 316.
47. Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., and Thornton, J. M. (1997). *Structure*, **5**, 1093.
48. Marchler-Bauer, A., Addess, K. J., Chappey, C., Geer, L., Madej, T., Matsuo, Y., *et al.* (1999). *Nucleic Acids Res.*, **27**, 240.
49. Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. (1995). *J. Mol. Biol.*, **247**, 536.
50. Rufino, S. D. and Blundell, T. L. (1994). *J. Comput. Aided Mol. Des.*, **8**, 5.

51. Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. J. Jr, Brice, M. D., Rodgers, J. K., *et al.* (1977). *J. Mol. Biol.*, **112**, 535.

52. Kraulis, P. J. (1991). *J. Appl. Crystallogr.*, **24**, 946.