Protein Structure Alignment

William R. Taylor and Christine A. Orengo

Laboratory of Mathematical Biology
The National Institute for Medical Research
The Ridgeway, Mill Hill, London NW7 1AA, U.K.

(Received 11 August 1988, and in revised form 6 January 1989)

A new method of comparing protein structures is described, based on distance plot analysis. It is relatively insensitive to insertions and deletions in sequence and is tolerant of the displacement of equivalent substructures between the two molecules being compared. When presented with the co-ordinate sets of two structures, the method will produce automatically an alignment of their sequences based on structural criteria. The method uses the dynamic programming optimization technique, which is widely used in the comparison of protein sequences and thus unifies the techniques of protein structure and sequence comparison. Typical structure comparison problems were examined and the results of the new method compared to the published results obtained using conventional methods. In most examples, the new method produced a result that was equivalent, and in some cases superior, to those reported in the literature.

1. Introduction

The comparison of protein structures has played an important role in developing our current understanding of protein structure and function. Through this approach, many duplicated domains and structural similarities have been identified, even between proteins with no apparent sequence identity. Comparison of more closely related structures has also proved useful in understanding how proteins accommodate slight alterations in amino acid sequence and the analysis of such changes provides a vital guide to the introduction of genetically engineered changes in sequence.

Structures have been compared by finding the superposition that will produce the closest approach between equivalent atoms and the technique of least-squares has generally been employed to find a best solution (for a review, see Matthews & Rossmann, 1985). This approach has two inherent limitations that become increasingly severe as the structures become less similar. Firstly, the equivalence of positions must be established before the superposition is performed and, secondly, the displacement of a subdomain within one structure can result in a poor overall fit between topologically equivalent structures.

In the many years that structure comparison has been practised, no wholly satisfactory solution has been found to either problem. They have been circumvented by an iterative approach in which the most obviously equivalent features between two structures are initially identified. This is often determined by visual inspection (usually with the aid of interactive computer graphics). However, the method of Rossmann and co-workers (Rao & Rossmann, 1973; Rossmann & Argos, 1975, 1976, 1977) and that of Remington & Matthews (1978, 1980) provide practical solutions to the problem. The former comprises a search in rotational space (comparing the structures in all possible orientations), while the latter is based on the comparison of all pairs of structural fragments between two structures. Both methods are computationally demanding and the latter method especially is sensitive to insertions and deletions between the sequences of the proteins compared.

Superposition based on an initial equivalence can be refined, as further equivalent positions may be identified once the structures are roughly superposed. New equivalent positions can again be found by "eye" and by algorithms described in the above works or by simpler algorithms described by Chothia & Lesk (1986) and Hubbard & Blundell (1987). The additional positions contribute to a new superposition and the process is repeated until no new equivalences are found. If, at any stage, it became apparent that there was a concerted displacement of a substructure relative to its topological equivalent structure then the only solution available in this methodology is to superpose the subfragments independently.

The distance, or Phillips (1970), plot is a useful representation of protein structure in the initial stages of determining whether there is any structural similarity between two proteins. This technique reveals both local patterns characteristic of secondary structure as well as their long-range

0022-2836/89/130001-22 $03.00/0

interactions, and equivalent features in different proteins can often be easily identified from a comparison of their plots. Attempts to automate the comparison of distance plots have been made, on the basis of matching the smaller plot (or matrix) with submatrices of the same size in the larger plot. Solutions can be found if the larger plot contains a domain that corresponds well to the smaller protein (see, for example, Padlan & Davies, 1975) or if the smaller plot represents only a protein fragment in which insertions and deletions are not expected (see, for example, Jones & Thirup, 1986). In general, these methods are severely restricted by insertions and deletions of sequence between the two proteins compared. The problem of insertions was treated by Nishikawa & Ooi (1974) by using the sequence alignment as a guide to structural equivalence but this is limited to situations where the sequences can be confidently aligned. The more abstract comparison method of Kuntz (1975) is not so sensitive to variation in the protein sequence length but can be used only to identify relatively local supersecondary structure. Richards & Kundrot (1988) described a method combining aspects of some of these approaches in which small distance masks, representing secondary structures, were matched against the distance plot of an intact structure.

In this paper, we describe a new method of comparing protein structures, based on distance plot analysis, that is completely general and relatively insensitive to insertions and deletions. The method requires no initial "seeding" of the structural equivalence and when presented with the coordinate sets of two structures will produce an alignment of their sequences based on structural criteria. The method uses the dynamic programming optimization technique, which is widely used in the comparison of protein sequences (Needleman & Wunsch, 1970), and thus unifies the techniques of protein structure and sequence comparison.

2. Methods

In the comparison of protein sequences, a matrix defining relatedness between the different amino acid types is used. Typically, this is the matrix of Dayhoff *et al.* (1978), which provides a measure of similarity between pairs of amino acid residues in the two sequences being compared. The dynamic programming method (see below and Fig. 1) can then be used to find the best alignment of the two sequences from a matrix defined by the comparison of all pairs of positions between the two sequences (usually referred to as the score matrix in the sequence alignment field).

Instead of sequences, if two structures were compared that had been previously superposed, the same technique could use a "real" (3-dimensional) distance between positions to determine a sequence alignment. However, the problem we address below is to compare independent structures without first performing a spatial superposition. To do this it is necessary first to develop a distance measure that is not dependent on their co-ordinate reference frame.

The distance plot provides an ideal representation in which to approach this problem, as it is independent of the co-ordinate frame and contains almost all the information of the Cartesian representation (only chirality is lost). To compare residues in different structures, we used the distance of a given residue to all other residues in the same structure. This defines a structural environment for the residue that can be expected to remain constant between equivalent positions in different structures and is invariant under rotation. The essence of our method is the development of an algorithm to compare these structural environments and to define a rigorous measure of similarity between them. We will first describe a basic method and proceed through progressive refinements, as this should produce a clearer description and also reflects the development of our work.

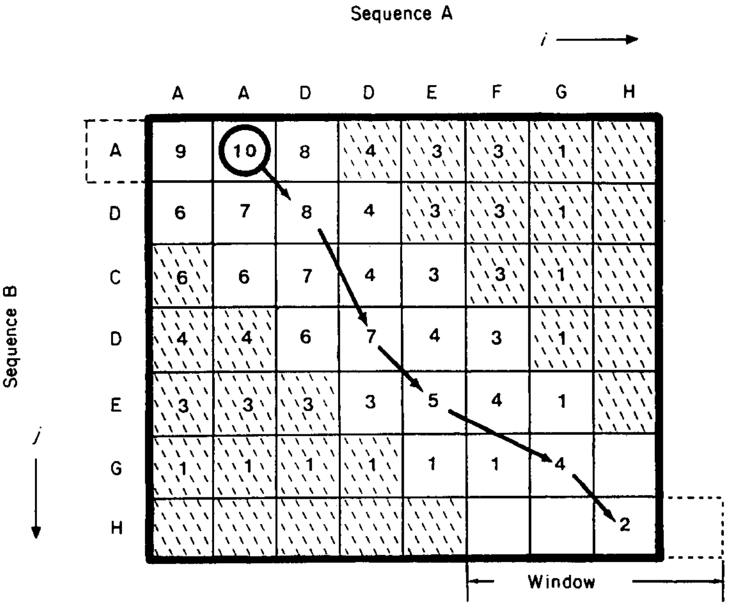
(a) *The dynamic programming algorithm*

The dynamic programming method is better known to biologists as the Needleman & Wunsch (1970) algorithm, as this was its first application to a biological problem. (For a wider review of biological applications, see Sankof & Kruskal, 1983.) It is a very general method used to find the optimal alignment of two linear sequences, given a definition of a relationship between the elements composing the sequences. The definition of the relationship may either be generic, in which every element of a given type has a fixed relationship to every other type (as in biological sequence alignment), or the relationship may be unique to every pair of positions.

In this paper we describe the application of this algorithm to the comparison of three-dimensional objects rather than the one-dimensional sequences to which it has been confined previously. As our application bridges the fields of sequence and structure analysis and familiarity with the algorithm is crucial for an understanding of our work, we have provided a simple worked example in the legend to Figure 1 that illustrates the alignment of two short sequences. Beginning with the C termini of the sequences (at the lower right of the matrix), scores are accumulated towards the beginning of the sequences (upper left). The highest score (S) is then found and from its location, a path is traced back by reversing the route by which the values leading to that score were originally accumulated. Mathematically, the algorithm is defined by the following recurrence equation:

$$S_{ij} = D_{ij} + \max \{ \begin{array}{ll} & S_{i+1,j+1}; \\ \max\limits_{k=i+2\to N_A} & S_{k,j+1} - g; \\ \max\limits_{l=j+2\to N_B} & S_{i+1,l} - g; \} \end{array} \qquad (1)$$

Where S is any element in the score matrix, D is a measure of relatedness between members of the sequence (e.g. Dayhoff's matrix) and g is a constant



	A	A	D	D	E	F	G	H
A	9	10	8	4	3	3	1	
D	6	7	8	4	3	3	1	
C	6	6	7	4	3	3	1	
D	4	4	6	7	4	3	1	
E	3	3	3	3	5	4	1	
G	1	1	1	1	1	1	4	
H								2

AAD-DEFGH
ADCDE-GH

Figure 1. The basic dynamic programming algorithm. The dynamic programming algorithm is illustrated on the alignment of 2 short sequences (AADDEFG with ADCDEGH) using a simple scoring scheme. Beginning at the lower right of the matrix, scores are accumulated towards the upper left. The highest score is then found and from its location, a path is traced back by reversing the route by which the values leading to that score were originally accumulated.

Matching identical characters scores 2 and the insertion of a gap costs 1. Beginning at the ends of the sequences (bottom right of the matrix) the score of 2 is entered in the cell cross-referenced by the matching Hs. Sums of the scores are accumulated working back towards the start of the sequences. Thus, the cell corresponding to the aligned Gs scores 2 and inherits 2 from the matched Hs, giving 4. None of the other positions on the same row or column has any matches but they still inherit 2 from the matched Hs less 1 for the insertion of a gap (giving rise to lines of 1s). The next diagonal position (EF match) has no score but inherits the preceding score of 4. However, an insert of a gap allows Es to match gaining 1 (2 less the gap penalty). This procedure (defined by eqn. (1)) is repeated for every cell until the matrix is complete. The highest score is then found (circled 10) and its inheritance path is traced back towards the opposite corner of the matrix (arrows). The corresponding alignment of the 2 sequences is shown below the matrix. This simple process has the remarkable property that it guarantees to find the best alignment for the given constraints.

The window parameter with a value of 4 instructs the program to ignore the shaded cells and thus reduce calculation time.

gap penalty. The lengths of the sequences A and B are N_{A} and N_{B}, respectively. For computation efficiency the maximum row and column values are stored and not calculated at every step (for a review, see Waterman, 1984). In the text, the result of recursive application of the above equation is represented simply as: $S = \max\{\ldots\}$.

(b) *Basic interatomic distance matching method*

In its simplest formulation the method considers only the α-carbon atoms of two structures and compares the distances between them. A measure of similarity between two pairs of positions requires that similar distances score highly so the simple form of a reciprocal difference of length was used as follows:

$$s = a/(|{}^{\mathrm{A}}d_{ij} - {}^{\mathrm{B}}d_{kl}| + b), \tag{2}$$

where s is the score obtained in comparing the distance, d, between atoms i and j in protein A with the distance between atoms k and l in protein B. The modulus of the difference is, of course, taken and the addition of the constant b moderates high scores (preventing division by zero). The constant a limits the maximum possible score. In the comparison of two positions, an ensemble of distances will be compared by the above measure and the overall score given by the sum of individual comparisons (s) as follows:

$$S_{ik} = \sum_{m=-n}^{+n} a/(|{}^{\mathrm{A}}d_{i,i+m} - {}^{\mathrm{B}}d_{k,k+m}| + b). \tag{3}$$

The number of distances compared is $2n$, with each comparison centred on atoms i and k in structures A and B, respectively, giving an overall score S_{ik} for the comparison of the two positions. Having defined a score between positions, the matrix of the scores of all pairs of positions between the structures can then be processed by a standard dynamic programming algorithm to find the best alignment of the positions (see above and Fig. 1).

As an alternative distance measure, the separation of β-carbons was used (for glycine the position of a dummy β-carbon atom was calculated). This measure increased the distinction between residues, especially those on either side of a β-sheet, by increasing the difference in separation from the point of view of residues above and below the sheet.

(c) *Structural environment matching by dynamic programming*

Except for the dynamic programming solution of the best equivalence of positions, it will be apparent that the above method is similar to the matrix-matching methods mentioned in Introduction and will encounter similar difficulties in the vicinity of insertions and deletions of sequence. This basic method was found to be adequate for matching local structures (i.e. n is small in eqn (3)) but was disrupted if the range of comparison ($-n$ to $+n$) spanned an insertion/deletion discontinuity between equivalent positions.

A solution to this problem was found by applying the dynamic programming algorithm to the lower level of distance comparison to produce a best equivalence of positions between the two environments being compared. Equation (3) thus becomes:

$$S_{ik} = \max\{a/(|{}^{A}d_{ij} - {}^{B}d_{kl}| + b)\}, \tag{4}$$

where S_{ik} is the maximum score obtained by the dynamic programming algorithm operating on the rectangular matrix defined by all values of j over the length of protein A and all values of l over the length of protein B. (The notation max $\{\ldots\}$ represents the application of eqn (1).) S_{ik}, as before, is an element in the higher (residue) level score matrix that comprises all values of S for i and k over the lengths of the sequences of the proteins (A and B) and is processed by the same dynamic programming algorithm. This formulation of the method was found to be tolerant of insertions and deletions, enabling sequentially remote residues to contribute constructively to the alignment.

A value of 50 was found to be suitable for the constant a and small values (up to 5) were tried for b.

(d) *Alignment dependency between levels*

Every comparison of positions using equation (4) will produce an alignment of matched distances, which is in itself an alignment of the two sequences. If only the maximum score (S_{ik}) is entered in the higher (residue) level matrix, the information contained in the lower (distance) level alignment will be lost. However, this information could be used to help determine the alignment at the higher level. To effect such a bias, all the values along the trace-back path in the lower level matrix were accumulated in corresponding elements in the higher level matrix (see Fig. 2). This produces a consensus of alternative alignments in the upper matrix, the best of which is, as before, identified by the standard dynamic programming algorithm. The values passed back from the lower alignment are equivalent to s, defined in equation (2). This has the advantage that large contributions are made only to the upper matrix for regions that match well in the lower comparison (typically, those that are physically adjacent to the positions compared).

In any comparison, the number of unequivalent positions compared will greatly outnumber the equivalent positions (typically, in the ratio $N \times M : N$, where N and M are the sequence lengths, with N the shorter). Since there is a general background or random score associated with the comparison of dissimilar positions, it was found necessary to introduce a cutoff on S to prevent the excessive accumulation of background noise. The value of the cutoff should increase with protein size but not linearly as large proteins tend to be composed of subdomains. Empirically, the relationship $(200 \times N)^{1/2}$ was found to be suitable, where N is the length of the shorter sequence.

The incorporation of these features greatly sharpened the resolution of the preferred path (or alignment) in the score matrix.

(e) *Vector comparison method*

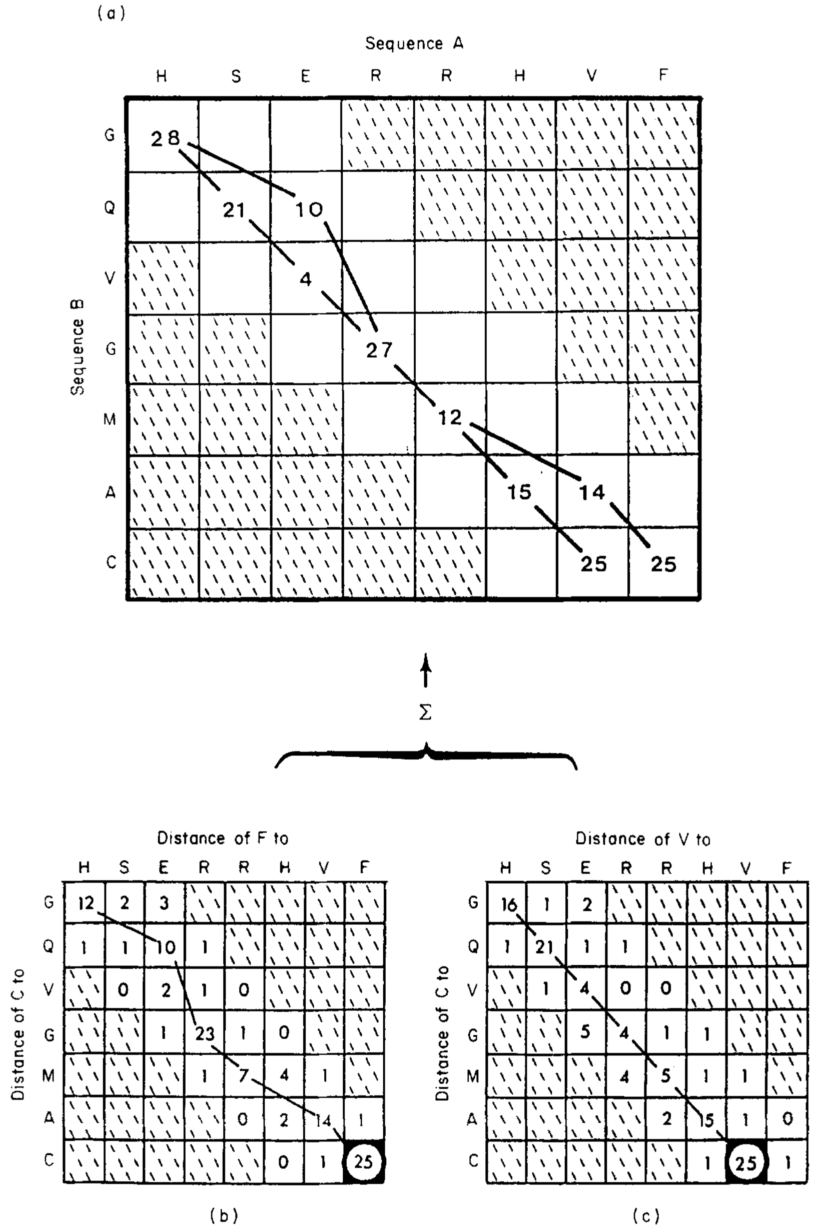
The above method, which is based solely on interatomic distances, has the limitation that similar distances achieve a high score even when these are between pairs of atoms that might be in completely different relative directions. To avoid this loss of potentially powerful directional information, the method was extended to allow the comparison of interatomic vectors rather than simple distances. For this approach to recognize positions in different orientations between structures in different co-ordinate frames, the interatomic vectors must be defined in a local frame of reference for every residue. Such a co-ordinate frame was readily available for each residue from the geometry of the α-carbon atom. A local X-axis was defined by the N–C vector and a tentative Y-axis by the C^{β}–H vector. (A dummy hydrogen position was calculated with a C^{α}–H bond length equal to the C^{α}–C^{β} bond.) The Z-axis was defined as their mutual perpendicular and the Y-axis redefined as perpendicular to X and Z to ensure orthogonality. A dummy β-carbon atom was used for glycine.

The vector equivalent of equation (2) which defines the comparison of distances could be identical in form (but instead calculating the modulus of the vector difference). However, as square roots are computationally expensive to calculate, the squared difference was used as follows:

$$s = a/(({}^{A}\mathbf{V}_{ij} - {}^{B}\mathbf{V}_{kl})^2 + b), \tag{5}$$

where $\mathbf{V}$ is an interatomic vector. The constant a was maintained at 50·0 and a value of 2·0 was used for b as lower values seemed to place too great an emphasis on local geometry.

This measure was clearly superior to a simple distance comparison and was used to generate the results discussed below.



(a)

	H	S	E	R	R	H	V	F
G	28			▨	▨	▨	▨	▨
Q		21	10		▨	▨	▨	▨
V	▨		4			▨	▨	▨
G	▨	▨		27			▨	▨
M	▨	▨	▨		12			▨
A	▨	▨	▨	▨		15	14	
C	▨	▨	▨	▨	▨		25	25

(b)

	H	S	E	R	R	H	V	F
G	12	2	3	▨	▨	▨	▨	▨
Q	1	1	10	1	▨	▨	▨	▨
V	▨	0	2	1	0	▨	▨	▨
G	▨	▨	1	23	1	0	▨	▨
M	▨	▨	▨	1	7	4	1	▨
A	▨	▨	▨	▨	0	2	14	1
C	▨	▨	▨	▨	▨	0	1	25

(c)

	H	S	E	R	R	H	V	F
G	16	1	2	▨	▨	▨	▨	▨
Q	1	21	1	1	▨	▨	▨	▨
V	▨	1	4	0	0	▨	▨	▨
G	▨	▨	5	4	1	1	▨	▨
M	▨	▨	▨	4	5	1	1	▨
A	▨	▨	▨	▨	2	15	1	0
C	▨	▨	▨	▨	▨	1	25	1

Figure 2. Application of the dynamic programming method for matching structure. The method is used at 2 levels of comparison. First, to find the best equivalence of distances for the 2 residues being compared, then at a higher level to find the best equivalence of residues within 2 sequences being compared. (a) The score matrix between 2 peptides HSERRHVF and GQVGMAC. (b) The score matrix for comparison of all distances centred on residue C in sequence B with all distances centred on residue F in sequence A. The dynamic programming algorithm is used to find the best pathway through this matrix and the values along this path are then accumulated in the corresponding cells of the higher level matrix (a). (c) The lower level process is repeated for residues C (sequence B) and V (sequence A). When all residue pairs have been compared in this way and the values accumulated in matrix (a), the dynamic programming algorithm is then used to find the best pathway through matrix (a).

(f) *Gap penalty*

In common with sequence alignment, the method described above can incorporate a penalty for gap insertion. This is often a critical parameter in sequence alignment where many alternative alignments may be distinguished only by the size of the penalty. We have found the results described below to be relatively insensitive to a gap penalty and a token penalty of 5 was applied to a gap of any size at both levels of comparison.

(g) *General method*

The vector-based method uses effectively three "distance" plots, one in each dimension, and produces a combined distance measure from all three plots. There is no inherent reason why this approach should be limited to three dimensions, although additional spatial dimensions may be of little value in the study of proteins. However, higher dimensions could incorporate any data that can be defined at the residue level. The most obvious and useful measure is the nature of the amino acid residue itself. This contribution was incorporated as a weighted component from the matrix of Dayoff *et al.* (1978) as follows:

$$S_{ik} = \max\{(wD_{\mathrm{R}i\mathrm{R}k}+a)/(({}^{\mathrm{A}}\mathbf{V}_{ij}-{}^{\mathrm{B}}\mathbf{V}_{kl})^2+b)\}, \quad (6)$$

where D_{XY} is the value specified in the Dayhoff *et al.* (1978) matrix for the exchange of amino acids of type X and Y, which above are Ri and Rk, and w is a weight to determine the relative contributions of sequence and structure. A default value of 1·0 was assigned to w and the constant a was reduced to 40 to maintain an equivalence of good scores with previous formulations.

Inclusion of a sequence component in the match, which could be made at either or both of the comparison levels discussed above, effectively unifies the comparison of sequence and structure, allowing both to influence the alignment simultaneously. Other potentially useful measures include the comparison of solvent accessibility, hydrogen bonds and conformational angles. The incorporation of these aspects and the optimization of their relative weights will be described elsewhere (Taylor & Orengo, 1989).

(h) *Implementation*

The method described above is implemented as a computer program referred to as SSAP (structure and sequence alignment program). It can be optionally configured to align on either structure or sequence or both and can base the structure alignment on either scalar or vector difference distances as described above.

The implementation of the dynamic programming algorithm was the same as that used by Taylor (1989) for sequence alignment and incorporates a windowing feature to avoid the calculation of remote off-diagonal elements (see Fig. 1). Results in the form of a sequence alignment can be used to provide the position equivalences for an existing least-squares superposition program to produce a graphical display of the compared structures. The program was written in *C* and run on a microVAX-II under VMS. Typical execution times are given in Table 1. The geometric calculations were performed using variables of type float. In *C*, operations on this type are slow and the program is being changed to use integer arithmetic, which, it is estimated, should produce a significant increase in speed.

Table 1
Execution times for structural alignment program

Sequence length	Window size	c.p.u. time (min)
50	20	5·2
	30	10·3
100	30	36·0
	60	121·0
150	20	36·2
	40	130·4
	60	269·6

Summary of central processor unit (c.p.u.) times on a microVAX-II obtained using the structural alignment program, with sequences of varying length and different window settings.

(i) *Structural data*

A separate FORTRAN program was written to prepare the input data for SSAP. This processes a Protein Structure Data Bank entry (Bernstein *et al.*, 1977) to produce a file containing the atoms whose positions are to be compared (either α-carbon or β-carbon) and the local co-ordinate frame of the residue (as a 3 × 3 matrix). This program has been adapted to determine also a reference frame when only α-carbon positions are available with the X-axis set by the vector $\mathrm{C}^{\alpha}_{i-1}$-$\mathrm{C}^{\alpha}_{i+1}$. The program also extracts the secondary structure definitions.

All co-ordinate data sets were extracted from the Protein Structure Data Bank (Bernstein *et al.*, 1977) and include the following proteins (with their databank identifier in brackets). Azurin [1AZU] (this data set is missing all atoms up to and including the nitrogen of Cys3; so that this residue could be used, its nitrogen atom position was calculated from the C, CB and CA positions); haemoglobin [4HHB], immunoglobulin FAB (NEW) [3FAB]; immunoglobulin FC [1FC1]; intestinal calcium binding protein [2ICB]; lysozyme (hen egg-white) [6LYZ]; lysozyme (T4) [1LZM]; myoglobin [3MBN]; parvalbumin (carp) [3CPV]; plastocyanin [3PCY]; and rhodanese [1RHD].

3. Results

We applied our method to typical structure comparison problems that have been well characterized by other methods. These were chosen to cover a range of structural types and difficulty and are presented approximately in order of increasing difficulty.

(a) *Globins*

The structural correspondences among the globins have been exhaustively analysed by Lesk & Chothia (1980). As an example of our method applied to these classic all-α structures, we aligned the α and β-subunits of haemoglobin with each other and with myoglobin. The alignments obtained were identical with those determined by Lesk & Chothia (1980) for the core regions, with the exception of a few minor displacements at the end of some helices in the myoglobin/haemoglobin comparisons (Fig. 3).

A Table of their relateness as measured by our method is given in Table 2 and compared to equivalent r.m.s.† measures of similarity obtained by Lesk

† Abbreviations used: r.m.s., root-mean square; LYZ, hen-egg lysozyme; LZM, phage T4 lysozyme; PLY, plastocyanin; AZU, azurin; H and L, immunoglobulin heavy and light chains; C and V, constant and variable regions of immunoglobulins H and L.

Table 2

Structural comparison of the globins

HBA	100·91	84·66
1·38	HBB	97·54
1·43	1·50	MBN

Comparison of the values obtained for alignment of the globins using our method with those obtained by Lesk & Chothia (1980) using conventional superposition. Values in the upper right-hand triangle are the maximum scores (/1000) returned by our method. The r.m.s. differences measured by Lesk & Chothia are shown in the lower left-hand of the triangle. Their values are calculated only over core residues and also for theoretical reasons should not be expected to scale directly to the alignment scores (for explanation see the text). HBA, haemoglobin α-chain; HBB, haemoglobin β-chain; MBN, myoglobin.

& Chothia (1980) by conventional superposition. Both methods agree that the α and β-chains of haemoglobin are the most similar but our method finds the haemoglobin β-chain to be more similar than the α-chain to myoglobin. This difference might be accounted for by the difference in measures, since the r.m.s. measure is normalized for a number of atoms while our score is not. Furthermore, Lesk & Chothia calculated the r.m.s. difference over only the core residues, while we considered all residues. Nevertheless, it is worth noting that, theoretically, only a rough agreement can be expected between our score and the r.m.s. error. As a simple example, consider a bi-lobed structure that can undergo a hinge-bending conformational change. Compared to itself, the unbent form will have a zero r.m.s. error and our comparison score will be high. Comparing the bent to the unbent, however, might give a large r.m.s. value but because the local geometry in each domain is unchanged, our score would drop only slightly.

(b) *Calcium-binding proteins*

The EF-hand structure is a recurrent calcium binding motif typically associated with the calmodulin superfamily (Kretsinger, 1980). The motif consists of two α-helices flanking a loop that chelates the ion and two motifs normally constitute a domain. The structure was first identified in parvalbumin (Moews & Kretsinger, 1975) with helices C and D and (the eponymous) E and F forming motifs. Parvalbumin also contains a third

Table 3

Immunoglobulin constant/variable domain comparisons

		Constant domains			
		CL	CH1	CH2	CH3
	b	B C E F G H	B C E F G H	B C E F G H	B C E F G H
Variable domains VL	seq	- - - - - -	- - E F - -	- - - E F -	- - E - - -
	2	**- - - - - -**	**- C E F - -**	**- - - E F -**	**- - E - - -**
	8	- - - - - -	- C E F - -	- - - E - -	- - E - - -
	16	- - - - - -	- C E F - -	- - - - - -	- - - - - -
	20	- - - - - -	- - E - - -	- - - - - -	- - - - - -
	25	- - - - - -	- - - - - -	- - - - - -	- - - - - -
		B C E F G H	B C E F G H	B C E F G H	B C E F G H
VH	seq	- - E F - -	- - E F - -	- - E F - -	- - E - - -
	2	**- - E F - -**	**- C E F - -**	**- - E F - -**	**- - E F - -**
	8	- - E F - -	- C E F - -	- - - - - -	- - E - - -
	16	- C E F - H	- C E F - H	- - - - - -	- C E F - H
	20	- C E F - H	- - - - - -	- - - - - -	- - - - - H
	25	B C E F G H	- - - - - -	- - - - - H	- - - - - H

Immunoglobulin core strand (B, C, E, F, G, H) alignments. Results are shown in matrix form, each cell of which contains a summary of the alignments for the cross-referenced pair under different conditions. The 2nd line in each cell (in bold) gives the results for the default values of the parameters a and b ($a = 50$, $b = 2$, which is equivalent to $a = 100$, $b = 4$). The upper line shows the effect of introducing a sequence matching component, while the lower lines show the effect of increasing values of b. For definitions of abbreviations, see legend to Fig. 6.

```
      HBA         MBN         HBB

  1   V - V     1   V - H      2
  2   L + L     2   L * L      3
  3 a S = S a   3 a S + T a    4
  4 a P = E a   4 a E - P a    5
  5 a A + G a   5 a G = E a    6
  6 a D + E a   6 a E * E a    7
  7 a K * W a   7 a W + K a    8
  8 a T + Q a   8 a Q = S a    9
  9 a N * L a   9 a L + A a   10
 10 a V £ V a  10 a V £ V a   11
 11 a K + L a  11 a L + T a   12
 12 a A £ H a  12 a H + A a   13
 13 a A * V a  13 a V * L a   14
 14 a W £ W a  14 a W * W a   15
 15 a G + A a  15 a A + G a   16
 16 a K * K a  16 a K + K a   17
 17 a V * V a  17 a V + V a   18
 18 a G . E a  18 a E
 19   A . A    19   A
 20 a H - D a  20 a D + N a   19
 21 a A - V a  21 a V = V a   20
 22 a G = A a  22 a A = D a   21
 23 a E * G a  23 a G * E a   22
 24 a Y £ H a  24 a H £ V a   23
 25 a G * G a  25 a G * G a   24
 26 a A + Q a  26 a Q + G a   25
 27 a E @ D a  27 a D £ E a   26
 28 a A @ I a  28 a I @ A a   27
 29 a L @ L a  29 a L * L a   28
 30 a E £ I a  30 a I * G a   29
 31 a R @ R a  31 a R * R a   30
 32 a M @ L a  32 a L £ L a   31
 33 a F £ F a  33 a F * L a   32
 34 a L * K a  34 a K + V a   33
 35 a S £ S a  35 a S + V a   34
 36 a F £ H a  36 a H * Y a   35
 37 a P + P a  37 a P + P a   36
 38 a T * E a  38 a E + W a   37
 39 a T £ T a  39 a T * T a   38
 40 a K = L a  40 a L + Q a   39
 41 a T = E a  41 a E - R a   40
 42 a Y + K a  42 a K = F a   41
 43   F + F    43   F = F     42
 44   P . D    44   D . E     43
 45   H : R    45   R : S     44
 46   F - F    46   F : F     45
          K    47   K   G     46
 47   D   H    48   H . D     47
 48   L . L    49   L - L     48
          K    50   K - S     49
          T a  51 a T - T a   50
          E a  52 a E = P a   51
          A a  53 a A - D a   52
 49   S : E a  54 a E = A a   53
 50 a H . M a  55 a M = V a   54
 51 a G . K a  56 a K - M a   55
          A a  57 a A : G a   56
 52 a S = S a  58 a S = N a   57
 53 a A = E a  59 a E - P a   58
 54 a Q = D a  60 a D = K a   59
 55 a V + L a  61 a L = V a   60
 56 a K + K a  62 a K = K a   61
 57 a G + K a  63 a K = A a   62
 58 a H * H a  64 a H + H a   63
 59 a G * G a  65 a G + G a   64
 60 a K * V a  66 a V + K a   65
 61 a K * T a  67 a T * K a   66
 62 a V £ V a  68 a V @ V a   67
 63 a A @ L a  69 a L £ L a   68
 64 a D £ T a  70 a T * G a   69
 65 a A £ A a  71 a A £ A a   70
 66 a L @ L a  72 a L @ F a   71
 67 a T @ G a  73 a G * S a   72
 68 a N £ A a  74 a A * D a   73
 69 a A £ I a  75 a I £ G a   74
 70 a V + L a  76 a L * L a   75
 71 a A = K a  77 a K = A a   76
```

```
      HBA         MBN         HBB

 72   H + K    78   K + H     77
 73   V = K    79   K - L     78
 74   D . G    80   G . D     79
 75   D - H    81   H : N     80
 76   M + H    82   H - L     81
 77   P = E    83   E . K     82
 78   N = A    84   A : G     83
 79   A - E    85   E = T     84
 80 a L + L a  86 a L + F a   85
 81 a S = K a  87 a K = A a   86
 82 a A = P a  88 a P + T a   87
 83 a L + L a  89 a L + L a   88
 84 a S = A a  90 a A + S a   89
 85 a D - Q a  91 a Q + E a   90
 86 a L = S a  92 a S + L a   91
 87 a H + H a  93 a H * H a   92
 88 a A - A a  94 a A + C a   93
 89   H - T a  95 a T + D     94
 90   K - K    96   K = K     95
 91   L + H    97   H + L     96
 92   R - K    98   K + H     97
 93   V + I    99   I * V     98
 94 a D = P a 100 a P + D a   99
 95 a P = I a 101 a I + P a  100
 96 a V = K a 102 a K + E a  101
 97 a N * Y a 103 a Y + N a  102
 98 a F * L a 104 a L £ F a  103
 99 a K * E a 105 a E * R a  104
100 a L @ F a 106 a F £ L a  105
101 a L @ I a 107 a I @ L a  106
102 a S £ S a 108 a S £ G a  107
103 a H £ E a 109 a E £ N a  108
104 a C @ A a 110 a A £ V a  109
105 a L @ I a 111 a I @ L a  110
106 a L @ I a 112 a I @ V a  111
107 a V @ H a 113 a H * C a  112
108 a T £ V a 114 a V @ V a  113
109 a L @ L a 115 a L @ L a  114
110 a A * H a 116 a H * A a  115
111 a A £ S a 117 a S * H a  116
112 a H * R a 118 a R + H a  117
113   L * H   119   H * F    118
114   P + P   120   P + G    119
115   A - G   121   G : K    120
116   E = D   122   D + E    121
117   F * F   123   F * F    122
118 a T = G a 124 a G + T a  123
119 a P = A a 125 a A + P a  124
120 a A = D a 126 a D * P a  125
121 a V * A a 127 a A * V a  126
122 a H * Q a 128 a Q £ Q a  127
123 a A * G a 129 a G * A a  128
124 a S £ A a 130 a A £ A a  129
125 a L @ M a 131 a M @ Y a  130
126 a D £ N a 132 a N £ Q a  131
127 a K * K a 133 a K £ K a  132
128 a F £ A a 134 a A £ V a  133
129 a L £ L a 135 a L £ V a  134
130 a A * E a 136 a E * A a  135
131 a S * L a 137 a L * G a  136
132 a V * F a 138 a F * V a  137
133 a S + R a 139 a R + A a  138
134 a T + K a 140 a K + N a  139
135 a V = D a 141 a D = A a  140
136 a L = I a 142 a I = L a  141
137 a T = A a 143 a A = A a  142
138 a S : A a 144 a A = H a  143
139   K : K a 145 a K - K    144
140   Y : Y a 146 a Y = Y    145
          K a 147 a K
          E a 148 a E
          L a 149 a L : H    146
141   R . G   150   G
          Y   151   Y
          Q   152   Q
          G   153   G
```

Fig. 3.

redundant motif (helices A and B) that does not bind Ca^{2+}. The intestinal calcium binding protein (Szebenyi & Moffat, 1985) contains the expected two motifs, the N terminus of which has some minor changes in its Ca^{2+}-binding loop.

Our method aligned the correct ion binding motifs in both structures (see Fig 4(a)), ignoring the redundant motif in parvalbumin. As expected, two insertions were placed in the first Ca^{2+}-binding loop of parvalbumin. The location of the first gap agrees with that of Gariepy & Hodges (1983), while the second is displaced by two residues and, interestingly, is adjacent to a proline residue that is unique to the corresponding motif in the intestinal protein. Comparison of the two motifs (using the computer graphics program QUANTA; R. Hubbard, unpublished results, © Polygon Corporation (1987)) revealed that our placement of the gap is structurally preferable (see Fig. 4(b)). Interestingly, the position of the ion-binding residues was maintained at the expense of the spatial equivalence of the two intervening residues (K54 and G18). This divergence is reflected in the alignment in Figure 4(a), where the absence of any symbol between the two residues indicates their lack of structural similarity.

(c) *Rhodanese*

The crystal structure of rhodanese is composed of two similar alternating β/α-type domains. Superposition of these domains by a conventional least-squares technique revealed that the cores of the domains are almost identical but the loop regions vary in length (Ploegman *et al.*, 1978). There is no significant sequence identity between the two halves. We aligned the structures of the two domains of rhodanese and the resulting alignment is compared to that determined by Ploegman *et al.* (1978) in Figure 5.

Our alignment is identical in all but minor aspects to that determined by conventional means. As in the globin comparisons the differences observed include single residue displacement at the ends of α-helices (see Fig. 5). It can be seen from the separation of the positions given by Ploegman *et al.* (Table 2 of Ploegman *et al.*, 1978) that these equivalences are poor fits (all over 3 Å (1 Å = 0·1 nm) deviation). A larger difference is seen at the C terminus of the domains where our alignment introduces a two-residue displacement relative to that of Ploegman *et al.* (1978). Investigation of the superposed structures on interactive computer graphics (using molecular comparison facilities in QUANTA) revealed that our alignment is plausible.

(d) *Immunoglobulin domains*

The immunoglobulin molecule heavy (H) and light (L) chains comprise six all-β domains, which are of two types, constant (C) and variable (V). By using combinations of these abbreviations, the domains can be referred to as VL, VH, CL and CH1, CH2 and CH3, where the number on the latter indicates their order on the heavy chain (see Fig. 6(a)). The structural interrelationships of these domains have been well characterized by many workers including Amzel & Poljak (1979), Lesk & Chothia (1982) and Taylor (1986). (For consistency in referring to the β-strands, the designations in the latter work will be used (see Fig. 6(b).) The correct alignment of the central strands can easily be identified from the equivalence of the conserved disulphide cysteines (in strands B and G) and a conserved tryptophan (in strand C). The edge strands are more variable, with both the D and D′ strands being deletable, while the G–H hairpin has a variable length loop and the H strand in the variable type domains includes a β-bulge (GGG or GQG). The E–F hairpin is more constant in structure within variable and constant domain types but between domain types a large displacement in sequence is required to obtain the correct structural equivalence of the E strands. Strands B, C, E, F, G and H (up to the β-bulge) will be referred to below as the core strands and their alignments are given in Figure 6(c).

We applied our method to the comparison of each domain against every other and compared our results to those obtained by Lesk & Chothia (1982). Our method produced the correct alignment of the core strands in nine out of the 15 comparisons. All domains of like type were correctly aligned but in the comparison of the constant with variable domains, a two residue displacement was observed in some strands. A summary of the observed shifts is shown in Table 3.

Errors might be expected in the comparison of the variable and constant domains as overall, substantial differences exist between their structures. However, the reason for the displacements can best be understood from an examination of the hydrogen-bonding maps of the domains (see Fig. 5 of Amzel & Poljak (1979) for the crystallographers' assignments, or Fig. 4 of Taylor (1986) for the automatic assignments by the method of Kabsch & Sander (1983)). It is clear from these diagrams that, relative to the two cysteines that contribute to the intermolecular disulphide link, the β-sheets extend in different directions. Referring to the direction of both cysteine-containing strands as up (as in Fig. 5

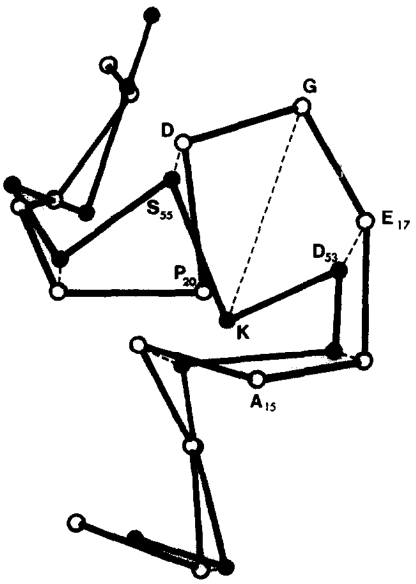
Figure 3. Alignment of myoglobin and haemoglobin chains. Alignments are shown in register for the comparison of the haemoglobin α-chain (HBA) with myoglobin (MBN) and for myoglobin with the haemoglobin β-chain (HBB). The sequences (in the 1-letter code) are flanked by a symbol denoting the secondary structure of the residue as defined in the Protein Structure Databank (a is α-helix; b is β-sheet) and their residue number. The symbol between 2 sequences is a measure of the strength of the alignment of the 2 adjacent residues. These are ranked in the following order beginning with a blank: " . : – + * £ @". The symbols linearly span the range between the weakest and the strongest similarity.

```
     CPV

    1   A
    2   F
    3   A
    4   G
    5   V
    6   L
    7 a N
    8 a D
    9 a A
   10 a D
   11 a I
   12 a A
A  13 a A
   14 a A
   15 a L
   16   E
   17   A
   18   C
   19   K
   20   A
   21   A
   22   D
   23   S
   24   F
   25   N
   26 a H
   27 a K
   28 a A
   29 a F
B  30 a F
   31 a A
   32 a K
   33 a V
   34   G
   35   L
   36   T
   37   S
```

```
     CPV       ICB

   38   K = K     1
   39   S = S a   2
   40 a A = P a   3
   41 a D - E a   4
   42 a D = E a   5
   43 a V + L a   6
   44 a K + K a   7
C  45 a K + G a   8
   46 a A + I a   9
   47 a F £ F a  10
   48 a A * E a  11
   49 a I + K a  12
   50 a I + Y a  13
   51 a D * A a  14
            A a  15
   52   Q * K a  16
   53 ● D . E ●  17
   54   K   G    18
   55 ● S : D ●  19
            P    20
   56   G + N    21
   57   F £ Q    22
   58   I £ L    23
   59 ● E * S ●  24
   60   E = K a  25
   61 ● D + E ●  26
   62   E £ E a  27
   63   L + L a  28
   64   K = K a  29
   65   L . L a  30
   66   F = L a  31
   67 a L . L a  32
   68 a Q - Q a  33
D  69 a N - T a  34
   70 a F - E a  35
   71 a K . F a  36
   72   A : P a  37
   73   D . S a  38
   74   A . L a  39
            L a  40
   75   R   K a  41
   76   A   G    42
            P    43
```

```
     CPV       ICB

   77   L : S    44
   78 a T : T a  45
   79 a D
   80 a G
   81 a E
   82 a T = L a  46
   83 a K = D a  47
E  84 a T = E a  48
   85 a F + L a  49
   86 a L * F a  50
   87 a K £ E a  51
   88 a A + E a  52
   89 a G £ L a  53
   90 ● D @ D ●  54
   91   S £ K    55
   92 ● D + N ●  56
   93   G * G    57
   94 ● D * D ●  58
   95   G + G    59
   96   K £ E    60
   97   I @ V    61
   98 ● G = S ●  62
   99   V + F a  63
  100 ● D * E ●  64
  101   E £ E a  65
  102 a F £ F a  66
  103 a T = Q a  67
  104 a A - V a  68
F 105 a L = L a  69
  106 a V - V a  70
            K a  71
            K a  72
            I a  73
  107 a K - S a  74
  108   A   Q a  75
```

(a)



(b)

Fig. 4.

RHDa				RHDb		
				P		149
1		V		A		150
2		H		I		151
3		Q		F		152
				K		153
				A		154
4		V		T		155
				L		156
5		L	.	N		157
6		Y		R		158
7		R	-	S		159
8	b	A	+	L	b	160
9	b	L	*	L	b	161
10	b	V	*	K	b	162
11	a	S	*	T	a	163
12	a	T	*	Y	a	164
13	a	K	+	E	a	165
14	a	W	*	Q	a	166
15	a	L	+	V	a	167
16	a	A	*	L	a	168
17	a	E	+	E	a	169
18	a	S	=	N	a	170
19	a	V	+	L	a	171
20	a	R	=	E	a	172
21	a	A	-	S	a	173
22	a	G				
23		K	:	K	a	174
24		V				
25		G				
26		P				
27	b	G	:	R		175
28	b	L	=	F		176
29	b	R	*	Q	b	177
30	b	V	£	L	b	178
31	b	L	£	V	b	179
32	b	D	@	D	b	180
33	b	A	*	S	b	181
34		S	+	R		182
35		W				
36		Y				
37		S				
38		P				
39		G				
40		T				
41		R				
42	a	E				
43	a	A	=	A	a	183
44	a	R	=	Q	a	184
45	a	K	=	G	a	185
46	a	E	=	R	a	186
47	a	Y	*	Y	a	187
48	a	L	=	L	a	188
49	a	E	-	G	a	189
				T		190
				Q		191
				P		192
				E		193
				P		194
				D		195
				A		196
				V		197
				G		198
				L		199
				D		200
				S		201
50	a	R	*	G		202
51		H	@	H		203
52		V	@	I		204
53		P	*	R		205
54		G	+	G		206
55		A	£	S		207
56	b	S	£	V	b	208
57	b	F	@	N	b	209
58	b	F	@	M	b	210
59		D	-	P		211
60		I	=	F		212
61		E	-	M		213
62		E	-	D		214
63		C	=	F		215
64		R	-	L		216
65		D	:	T		217
66		K	.	E		218
67		A				
68		S	:	N		219
69		P				
70		Y				
71		E	.	G		220
72		V				
73		M	-	F		221
74		L	=	E		222
75		P	*	K		223
76	a	S	*	S	a	224
77	a	E	+	P	a	225
78	a	A	+	E	a	226
79	a	G	£	E	a	227
80	a	F	*	L	a	228
81	a	A	*	R	a	229
82	a	D	*	A	a	230
83	a	Y	£	M	a	231
84	a	V	*	F	a	232
85	a	G	+	E	a	233
86	a	S	*	A	a	234
87	a	L	*	K	a	235
88		G	+	K		236
89		I	=	V		237
90		S	=	D		238
91		N	.	L		239
92		D	.	T		240
93		T	=	K		241
94	b	H	+	P	b	242
95	b	V	£	L	b	243
96	b	V	£	I	b	244
97	b	V	£	A	b	245
98	b	Y	*	T	b	246
99		N	£	C		247
100		G	:	R		248
101		D				
102		D				
103		L				
104		G	:	K		249
105		S	=	G		250
106		F	=	V	a	251
107	a	Y	*	T	a	252
108	a	A	@	A	a	253
109	a	P	£	C	a	254
110	a	R	*	H	a	255
111	a	V	*	I	a	256
112	a	W	+	A	a	257
113	a	W	=	L	a	258
114	a	M	£	A	a	259
115	a	F	£	A	a	260
116	a	R	*	Y	a	261
117	a	V	*	L	a	262
118	a	F	*	C	a	263
119	a	G	+	G	a	264
120		H	*	K		265
121		R	.	P		266
122	b	T	-	D		267
123	b	V	*	V		268
124	b	S	*	A	b	269
125	b	V	£	I	b	270
126	b	L	@	Y	b	271
127	b	N	=	D		272
128		G	*	G		273
129	a	G	£	S	a	274
130	a	F	@	W	a	275
131	a	R	£	F	a	276
132	a	N	*	E	a	277
133	a	W	@	W	a	278
134	a	L	£	F	a	279
135	a	K	+	H	a	280
136	a	E	*	R	a	281
137	a	G				
138		H	-	A	a	282
				P		283
				P		284
				E		285
139		P	-	T		286
140		V	=	W		287
141		T	=	V		288
142		S	.	S		289
143		E		Q		290
				G		291
				K		292
144		P		G		293
145		S				
146		R				
147		P				
148		E				

Figure 5. Alignment of the 2 α/β-type domains of rhodanese. The symbols and numbers are as defined in the legend to Fig. 3. RHDa and RHDb denote the 1st and 2nd domains, respectively, of rhodanese. In the core regions (β-strands and α-helices) the alignment corresponds to that determined by Ploegman *et al.* (1978) using conventional superposition methods.

Figure 4. (a) Alignment of parvalbumin with intestinal calcium-binding protein. The method correctly locates the corresponding motifs and aligns the residues involved in calcium binding (●). The sequences are annotated as described in the legend to Fig. 1. CPV, carp parvalbumin; ICB, intestinal Ca^{2+}-binding protein. The 3-dimensional structure of the loop between helices C and D in CPV is shown in (b).

(b) Superposition of Ca^{2+}-binding loops in CPV and ICB. α-Carbon virtual bonds are drawn between residues in carp parvalbumin and the intestinal calcium binding in the range of residues 48 to 60 and 11 to 25, respectively (corresponding to the CD hand in CPV, Fig. 4(a)). The α-carbons of CPV are shown by filled circles, while those of ICB are shown by open circles. The structures were superposed interactively using the computer graphics program QUANTA. Some equivalent atoms are connected by broken lines. The 2 insertions in the ICB structure (Ala15 and Pro20) can be seen clearly. The residues Glu17 and Asp19 in ICB and Asp53 and Ser55 in CPV provide co-ordination for their Ca^{2+} ions. The alignment in (a) correctly indicated very little similarity in the structural environments of Gly18 and Lys54.

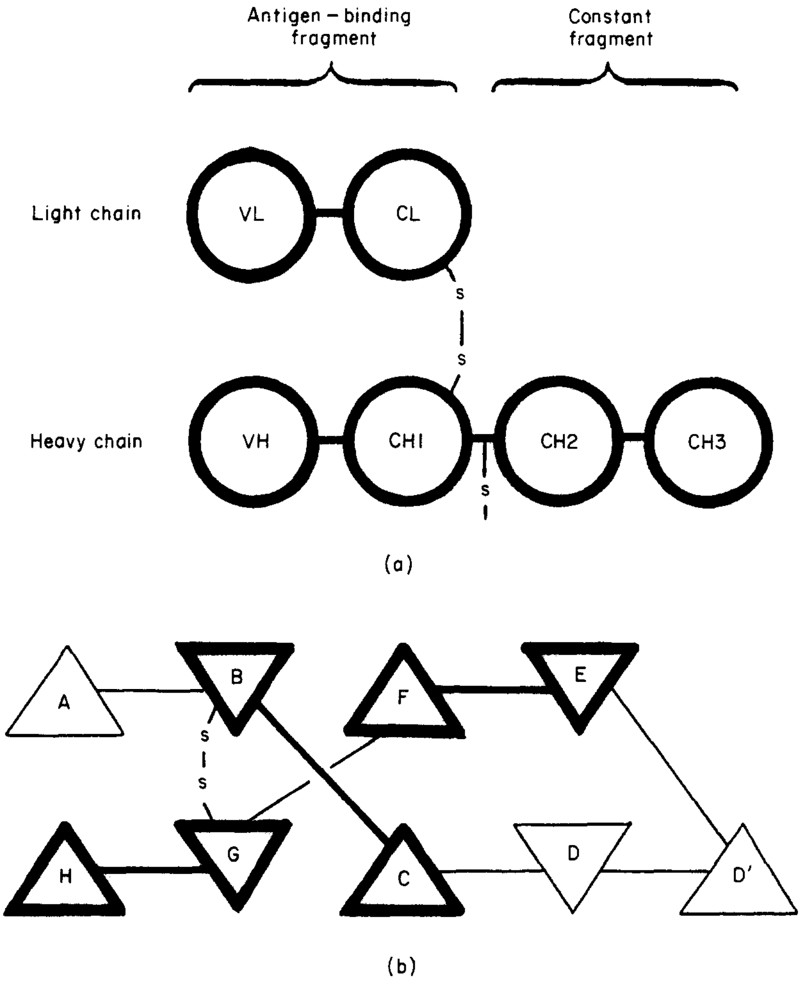


Figure 6. (a) Schematic outline of the domain structure of the immunoglobulin molecule. The prefixes V and C denote a variable or constant domain, respectively while the suffixes L and H denote the light and heavy chains, respectively. The various domains were extracted from the antigen-binding fragment (3FAB) and the constant fragment (1FC1). The conserved intermolecular disulphide bond between the chains is also shown.

(b) Schematic topology diagram for the structure of the immunoglobulin domains. A triangle indicates an approaching strand and an inverted triangle a receding strand. Strands are labelled sequentially such that each strand in every structure has the same designation as its topological equivalent in the CH2 structure. The strands common to all the structures are indicated by bold lines. The conserved intramolecular disulphide bond between strands B and G is indicated.

(c) Alignment of the immunoglobulin constant and variable domains. The alignments of the constant domains (CH1, CH2, CH3 and CL) with (1) the heavy-chain variable domain (VH) and (2) the light-chain variable domain (VL). The bars and colons beside the sequences indicate the core regions aligned by Lesk & Chothia (1980); they correspond to the β-strands indicated to the left. In addition, the alignment of the regions marked by bars were taken as the criteria for correctness in Table 3. The alignment of the marked regions is valid for all pairs of domains (including those of like types not shown). The alignment of the VH/CL domains is displaced by 2 residues in every strand, corresponding to a concerted displacement of one domain relative to the other. The only other errors result from a failure to allow for the β-bulge in the H-strand of the variable domains. Other symbols as in the legend to Fig. 3.

of Amzel & Poljak (1979)), the β-sheets in the variable domains both begin two residues lower than the constant domain and do not extend as high. Thus irrespective of a clear sequence identity, our program has attempted to equate the bulk of the β-sheets, producing some two residue displacements.

We reapplied our method to these comparisons with a bias for sequence matching (see Methods). With a sequence weight (w in eqn. (5)) of 2, all the conserved tryptophan residues in the C strand were correctly aligned but the E–F hairpin was still displaced by two residues in most comparisons (see Table 3). In this region the sequence alignment is very ambiguous and thus contributes little towards correcting the misplaced structures.

The relative size of the constants a and b in equation (3) (see Methods) determine the contrast between scores for well matched and poorly matched positions. It was expected that a reduced distinction between the degree to which individual positions correspond might be beneficial when comparing less similar structures. (The reasoning behind this can be found in Discussion.) Conse-

```
(I) VH domain
              /
                   CH1         CH2         CH3         CL

                      S                       R           A
                      T                       E           A
                  V = K       V + P       V + P       V * P
                  Q * G       Q + S       Q + Q       Q * S
                : L * P     : L £ V :   : L £ V :   : L £ V :
A strand        : E £ S     : E £ F :   : E £ Y :   : E @ T :
                : Q @ V :   : Q £ L :   : Q £ T :   : Q @ L :
                  S @ F :     S £ F       S £ L       S @ F
                  G * P :     G + P       G + P       G £ P
                  P = L       P * P       P * P       P + P
                  G . A       G . K       G   S       G . S
                  L = P           P       L   R       L : S
                  V - S           K           E       V   E
                  R - S           D
                  P   K       L : T
                  S . S       V . L       V   E           E
                  Q * T       R = M       R : M       R : L
                      S       P : I       P   T       P   Q
                      G       S . S       S   K       S . A
                      G       Q = R       Q = N       Q - N
                      T
                      A
                  T £ A |     T = T       T @ Q       T * K
                  L @ L |     L + P       L + V       L £ A
                | S £ G |   | S @ E |   | S @ S |   | S @ T |
                | L @ C |   | L @ V |   | L @ L |   | L @ L |
                | T @ L |   | T @ T |   | T @ T |   | T @ V |
B strand        | C @ V |   | C @ C |   | C @ C |   | C @ C |
                | T @ K     | T @ V |   | T @ L |   | T @ L |
                | V - D     | V @ V |   | V @ V |   | V @ I |
                  S = Y       S £ V       S £ K       S £ S
                  G : F       G + D       G + G       G * D
                  T           T = V       T * F       T * F
                  S           S - S       S - Y       S - Y
                                  H
                                  E
                  F           F : D       F   P       F   P
                  D           D £ P       D * S       D + G
                  D - P       D * Q       D * D       D @ A
                  Y - E       Y = V       Y £ I       Y £ V
                | Y @ P     | Y £ K |   | Y * A |   | Y * T |
                | S @ V     | S £ F |   | S @ V |   | S @ V |
                | T @ T |   | T £ N |   | T @ E |   | T @ A |
C strand        | W @ V |   | W £ W |   | W @ W |   | W @ W |
                | V * S |   | V £ Y |   | V @ E |   | V £ K |
                  R + W |     R * V       R £ S       R * A
                  Q : N |     Q - D       Q = N       Q £ D
                  P : S       P           P           P : S
                  P           P           P           P
                  G           G           G           G
                  R           R           R           R
                  G           G           G           G
                  L           L = G       L + G       L
                  E           E + V       E * Q       E + S
                  W           W * Q       W * P       W * P
D strand          I           I * V       I * E       I * V
                  G           G + H       G           G + K
                  Y           Y : N       Y           Y
                  V           V           V           V
                  F           F           F           F
                  Y           Y           Y           Y
                  H           H           H           H
                  G           G           G           G
                  T           T           T           T
                  S           S           S           S
                  D           D           D           D
                  T           T           T           T
                  D           D           D           D
                  T   G       T           T           T
```

(c)

Fig. 6.

```
               P : A        P            P            P
               L : L        L            L            L
               R : T        R            R            R
               S : S        S            S            S
               R + G |      R            R            R
               V + V |      V £ A        V = N        V + A
             | T = H |    | T £ K |    | T - N |    | T + G |
E strand     | M + T      | M @ T |    | M = Y |    | M + V |
             | L + F      | L @ K |    | L = K |    | L = E |
               V = P        V £ P        V - T        V - T
               N : A        N @ R        N + T        N = T
               T - V        T . E        T   P            T
               S . L        S + Q        S   P            P
               K + Q            Q        K - V        T . S
                                Y            L        S . K
                                N            D            Q
                                             S            S
                                             D            N
                   S            S            G            N
                   S            T            S            K
                   G            Y            F            Y
                   L        K £ R            F        K = A
             | N + Y      | N £ V |    | N * L |    | N + A |
             | Q @ S      | Q @ V |    | Q @ Y |    | Q @ S |
F strand     | F @ L |    | F @ S |    | F @ S |    | F @ S |
             | S @ S |    | S @ V |    | S @ K |    | S @ Y |
             | L @ S |    | L @ L |    | L @ L |    | L @ L |
             | R £ V |    | R @ T |    | R @ T |    | R £ S |
               L @ V |      L = V        L £ V        L £ L
               S £ T |      S - L        S - D        S + T
               S = V        S : H        S . K        S : P
               V : P        V = Q        V * S        V * E
                   S        T : N        T : R        T * Q
                   S        A   W        A   W        A   W
                   S
               T . L
               A . G
               A . T        A   L        A   Q        A : K
               D : Q        D - D        D - Q        D - S
               T £ T        T + G        T + G        T - H
               A - Y |      A * K        A * N        A = K
               V * I |      V * E        V @ V        V £ S
             | Y @ C |    | Y @ Y |    | Y @ F |    | Y @ Y |
             | Y * N |    | Y @ K |    | Y @ S |    | Y @ S |
G strand     | C £ V |    | C @ C |    | C @ C |    | C @ C |
             | A £ N |    | A @ K |    | A £ S |    | A @ Q |
             | R £ H      | R * V |    | R £ V |    | R * V |
             | N - K      | N - S |    | N = M |    | N - T |
               L - P        L . N        L . H        L . H
               I            I   K        I   E        I   E
               A            A   A        A   A        A
               G            G   L        G   L        G
               C            C   P        C   H        C
               I            I : A        I . N        I - G
             | D   S      | D = P |    | D : H      | D   S
H strand     | V + N      | V - I |    | V + Y |    | V * T |
             | W = T      | W * E |    | W = T |    | W - V |
             | G : K |    | G * K |    | G - Q |    | G = E |
β-bulge        Q = V |      Q - T :      Q + K |      Q = K |
             : G + D |    : G + I :    : G £ S :    : G * T :
             : S * K |    : S : S      : S £ L :    : S £ V :
               L £ K :      L = K        L * S        L * A
               V £ V :      V   A        V = L        V = P
               T * E        T . K        T   S        T - T
               V : P        V            V            V   E
               S   K        S            S            S   C
               S   S        S            S            S   S
               A - C        A            A            A
```

(c)

Fig. 6.

(2) VL domain

```
                 /
                   CH1          CH2          CH3          VL

                     K                         R            A
                     G                         E            A
                 S : P        S . P        S : P        S : P
                 V * S        V * S        V + Q        V * S
               : L * V :    : L @ V :    : L £ V :    : L * V :
A strand       : T * F :    : T £ F :    : T £ Y :    : T * T :
               : Q * P :    : Q @ L :    : Q £ T :    : Q * L :
                     L        P * F        P + L        P + F
                 P = A        P - P        P : P        P : P
                              S : P
                              V : K
                 P = P            P        S . P        S . P
                 S = S            K        V . S        V : S
                 V : S            D        S . R        S = S
                 S   K        S - T            E        G . E
                 G : S        G . L        G : E            E
                 A   T        A = M        A : M        A . L
                 P   S        P - I        P   T        P   Q
                 G   G        G - S        G . K        G . A
                 Q   G        Q = R        Q * N        Q + N
                 R : T        R * T        R £ Q        R £ K
                 V : A        V £ P        V @ V        V £ A
               | T * A |    | T £ E |    | T @ S |    | T @ T |
               | I * L |    | I @ V |    | I @ L |    | I @ L |
               | S * G |    | S £ T |    | S @ T |    | S £ V |
B strand       | C * C |    | C @ C |    | C @ C |    | C @ C |
               | T * L |    | T £ V |    | T @ L |    | T £ L |
               | G * V |    | G @ V |    | G @ V |    | G @ I |
                 S * K        S + V        S + K        S * S
                 S : D        S - D        S : G        S - D
                 S * Y        S : V        S £ F        S + F
                 N   F        N   S        N   Y        N   Y
                 I   P        I   H        I   P        I   P
                                  E
                 G : E        G   D        G : S        G
                 A            A : P        A : D        A - G
                 G * P        G = Q        G            G + A
                 N * V        N * V        N * I        N £ V
               | H * T |    | H @ K |    | H + A |    | H * T |
               | V * V |    | V @ F |    | V * V |    | V £ V |
C strand       | K * S |    | K @ N |    | K * E |    | K £ A |
               | W * W |    | W @ W |    | W £ W |    | W £ W |
               | Y = N |    | Y @ Y |    | Y * E |    | Y £ K |
                 Q            Q £ V        Q * S        Q * A
                 Q            Q + D        Q - N        Q + D
                 L            L            L            L
                 P            P            P            P
                 G            G            G            G
                 T            T            T            T
                 A            A            A            A
                 P            P - G        P : G        P . S
                 K : S        K * V        K = Q        K - S
                 L = G        L £ Q        L = P        L = P
D strand         L = A        L @ V        L = E        L = V
                 I : L        I * H        I . N        I - K
                 F * T        F £ N        F            F
                 H            H            H            H
                 N            N            N            N
                 N            N            N            N
                 A            A            A            A
                 R            R            R            R
                 F * S        F £ A        F            F + A
               | S * G |    | S £ K |    | S + N |    | S £ G |
E strand       | V * V |    | V @ T |    | V @ Y |    | V £ V |
               | S = H |    | S £ K |    | S £ K |    | S £ E |
                 K * T        K @ P        K @ T        K £ T
                 S * F        S * R        S £ T        S + T
                 G : P            E        G - P        G : T
                     A            Q            P            P
```

(c)

Fig. 6.

```
                      V            Q            V            S
                      L            Y            L            K
                      Q                         D            Q
                      S                         S            S
                      S            N            D            N
                      G            S            G            N
                      L            T            S            K
                      Y            Y            F            Y
                      S        G - R            F            A
                | S = L |    | S = V |    | S * L |    | S + A |
                | S * S |    | S £ V |    | S @ Y |    | S £ S |
F strand        | A @ S |    | A £ S |    | A @ S |    | A @ S |
                | T * V |    | T @ V |    | T @ K |    | T £ Y |
                | L * V |    | L @ L |    | L @ L |    | L @ L |
                | A * T |    | A £ T |    | A @ T |    | A £ S |
                  I * V        I + V        I @ V        I @ L
                  T : P        T : L        T = D        T + T
                  G            G . H        G : K        G : P
                  L = S        L - Q        L + S        L = E
                  Q = S        Q : N        Q : R        Q * Q
                  A : S        A . W        A   W        A   W
                  E   L        E   L        E   Q        E : K
                  D : G        D : D        D = Q        D : S
                  E * T        E = G        E + G        E . H
                  A * Q        A £ K        A £ N        A + K
                  D * T        D £ E        D £ V        D * S
                | Y * Y |    | Y @ Y |    | Y @ F |    | Y @ Y |
                | Y * I |    | Y @ K |    | Y @ S |    | Y @ S |
G strand        | C * C |    | C @ C |    | C @ C |    | C @ C |
                | Q * N |    | Q @ K |    | Q @ S |    | Q @ Q |
                | S * V |    | S @ V |    | S @ V |    | S @ V |
                | Y * N |    | Y £ S |    | Y @ M |    | Y £ T |
                  D = H        D = N        D £ H        D £ H
                      K        R - K        R £ E        R £ E
                                   A        S . A        S £ G
                      P                     L = L        L £ S
                  R * S            L            H
                  S * N        S : P            N
                  L * T        L = A            H
                | R * K |    | R = P |    | R * Y |    | R @ T |
                | V * V |    | V + I |    | V * T |    | V @ V |
H strand        | F * D |    | F + E |    | F * Q |    | F £ E |
                | G * K |    | G + K |    | G * K |    | G £ K |
β-bulge           G * K :      G = T :      G + S :      G + T :
                : G = V :    : G = I :    : G - L :    : G + V :
                : T = E      : T : S      : T : S      : T - A
                  K : P        K - K        K = L        K + P
                  L   K        L   A        L   S        L   T
                  T   S        T   K        T            T . E
                  V   C        V            V            V   C
                  L   C        L            L            L   S
                  R            R            R            R
```

(c)

Fig. 6.

quently, we attempted the constant/variable comparisons with an increased value of 4 for b. However, this led to a generally reduced score, which in some comparisons scored zero. To avoid this constant a was doubled to raise the score and b doubled again to maintain their desired ratio (i.e. $a = 100$, $b = 8$). This combination of parameters produced some significant improvements in the alignments, notably, those involving the CH2 domain. With a second doubling of the value of b (to 16), the alignments with the VL domain continued to improve but most of those involving the VH domain became worse. Strangely, a further increase in b largely reversed the latter trend, except in the VH/CL comparison. Neglecting this comparison for the moment, with a value of 20 for b, the resulting alignments of the other constant and variable domains continued to improve until, finally, with a value of 25 for b these were all correctly aligned in the core β-strand regions by the criteria of Lesk & Chothia (1980), except for minor shifts in the H strand associated with the β-bulge in the variable domain (Fig. 6(c)). The VH/VL comparison, however, had continued to worsen until, in the final run, all strands were displaced by two residues. Interestingly, this final alignment contained no internal discontinuities in the hydrogen-bonding pattern and constituted a shift of one intact protein relative to the other by two positions along the β-strands (see Table 3 for summary).

(e) *Lysozyme*

Comparison of the structures of a hen egg-white lysozyme (Blake *et al.*, 1965) and T4-phage lysozyme (Remington *et al.*, 1977) revealed a surprising equivalence (Rossmann & Argos, 1976, 1977; Matthews *et al.*, 1981), showing structural similarities in the small β-sheet and in the helices that surround it. T4 lysozyme, however, has no equivalent to the initial 25 residues in hen lysozyme but has over 60 additional residues in its C terminus. We used our method to repeat this comparison and compared the result principally to the alignment of Rossmann & Argos (1976), Weaver *et al.* (1985) and also to our own investigations using interactive computer graphics. Our results (Fig. 7(a)) agree in outline with those of Rossmann & Argos (1976), Matthews *et al.* (1981) and Weaver *et al.* (1985) in identifying both the large N and C-terminal displacements. In detail, however, significant differences were found.

We found the first common helix (residues 3 to 11 in LXM and 25 to 35 in LYZ) to be displaced by one residue from both published comparisons. However, from visual inspection of the superposed regions on computer graphics (using QUANTA), our alignment appears to be quite plausible and has the advantage of equating the functionally similar residues MLRID with AAKFE.

Our alignment of the initial β-strand agrees with that of Rossmann & Argos (1976), including the occurrences of a bulge, but our method excluded Ala42 (in LYZ) from the sheet, while Rossmann & Argos excluded Gln41. (Weaver *et al.* (1985) placed quite a different gap in LYZ opposite residues 20 and 21 in LZM.) The following β-meander (residues 43 to 60 in LYZ, 17 to 33 in LZM) seems to have been unnecessarily broken up by Rossmann & Argos and also by Weaver *et al.* Superposition of the β-sheets using the program QUANTA (Fig. 7(b)) supported the alignment in Figure 7(a), containing only a single gap, and confirms the placement of the gap in the second β-turn. The alignment of this region was also checked by comparing the pattern of hydrogen bonds as defined by the method of Kabsch & Sander (1983) (Fig. 7(c)). These data were also consistent with our alignment. The β-region includes (non-equivalent) residues in both structures that are involved in catalysis (Asp20 in LZM and Asp52 in LYZ). A greater degree of structural and sequence conservation might therefore be expected and the close similarity of KDTEGYYTIG to RNTDGSTDYG supports this.

The remainder of the alignment is the same for both methods, except for trivial displacements at the fringes of equivalent blocks. The C termini lie in roughly the same location but do not have any obviously equivalent structure. They were not aligned by Rossmann & Argos (1976) but Weaver *et al.* (1985) found some correspondence towards the C termini, as did our method.

(f) *Plastocyanin/azurin*

The small copper-binding proteins, plastocyanin (Garrett *et al.*, 1984) and azurin (Adman *et al.*, 1981) are all β-type structures that exhibit partial sequence and structural similarity. Each structure is composed of two sheets that stack together forming a β-sandwich. The problem of their superposition and the identification of equivalent residues has been studied by Chothia & Lesk (1982), who found it was difficult to obtain a good overall comparison of the two sheets as there is almost a 4 Å displacement between them. This superposition problem was also analysed by Adman (1985), who determined a different equivalence in one β-sheet. These alignments were compared with that determined by our method (Fig. 8(a)).

Except for some minor insertions and deletions on the fringes, our alignment agrees with the register determined by Chothia & Lesk (1982) and Adman (1985) for the second β-sheet (B in Fig. 8(a); sheet II in Fig. 2 of Chothia & Lesk (1982)). However, for sheet b our method produces a correspondence that agrees with Adman but is displaced by a register of two residues over the whole sheet compared to the Chothia & Lesk alignment. We investigated this difference using QUANTA and began by superposing sheet B on to the undisputed alignment of strands. We found that sheet b is then placed in a register intermediate between the Chothia & Lesk (1982) alignment and ours. As this alignment is out of step with the phase of residues on either side of the sheet, one of the alternative adjacent alignments must be chosen.

From the sequence some features suggested that our alignment (and that of Adman (1985)) was preferable. In the first strand in sheet b, we align IdVIL with VdIqG (in PCY and AZU, respectively: upper-case residues are internal), while Chothia & Lesk (1982) align IdVIL with CsVdI, both of which are functionally comparable but the former aligns with two aspartic acid residues. There is little to compare in the second strand in sheet b but in the third (residues 67 to 74 in PCY, 91 to 99 in AZU) we align EtFeVaL with EkDsVtF, giving two identities where before none was found, which agrees with the sequence alignment determined by Argos (1987). A structural superposition of the two sheets to illustrate our alignment is shown in Figure 8(b).

4. Discussion

The above results demonstrated that the method described here produced a structural equivalence between pairs of structures that was, in most examples, equal in quality, and in some cases superior, to those reported in the literature. Many of the structural comparisons considered above were originally

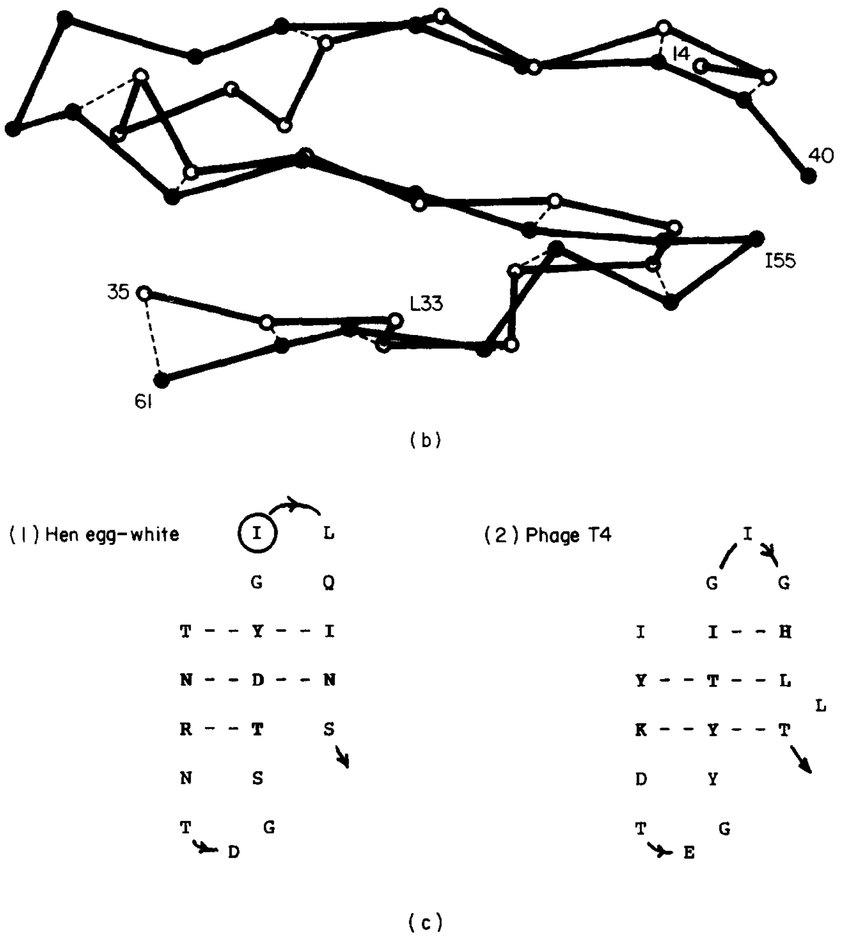
```
LZM         LYZ

            K b    1
            V b    2
            F b    3
            G      4
            R a    5
            C a    6
            E a    7
            L a    8
            A a    9
            A a   10
            A a   11
            M a   12
            K a   13
            R a   14
            H a   15
            G     16
            L     17
            D     18
            N     19
            Y     20
            R     21
 1     M .  G     22
            Y     23
 2     N -  S     24
            L a   25
            G a   26
 3 a   I -  N a   27
 4 a   F =  W a   28
            V a   29
 5 a   E +  C a   30
 6 a   M =  A a   31
 7 a   L =  A a   32
 8 a   R =  K a   33
 9 a   I =  F a   34
10 a   D +  E a   35
11 a   E =  S     36
12     G -  N     37
13     L .  F b   38
14 b   R .  N b   39
15 b   L :  T b   40
16 b   K :  Q     41
            A b   42
17 b   I -  T b   43
18 b   Y -  N b   44
19 b   K :  R b   45
20 b   D -  N b   46
21     T :  T     47
22     E =  D     48
23 b   G -  G     49
24 b   Y +  S b   50
25 b   Y £  T b   51
26 b   T £  D b   52
27 b   I *  Y b   53
28 b   G =  G b   54
            I     55
29     I £  L     56
30     G =  Q b   57
31 b   H @  I b   58
32 b   L *  N b   59
33 b   L -  S b   60
34 b   T    R     61
35     K    W     62
            W     63
36     S    C     64
37     P
38     S .  N     65
39 a   L    D     66
40 a   N    G     67
41 a   A    R     68
```

```
LZM         LYZ

            T      69
            P      70
            G      71
            S      72
            R      73
            N      74
            L      75
            C      76
            N      77
            I      78
            P      79
42 a   A :  C a    80
43 a   K :  S a    81
44 a   S :  A a    82
45 a   E :  L a    83
46 a   L :  L a    84
47 a   D .  S      85
48 a   K
49 a   A
50 a   I
51     G
52     R
53     N
54     C
55     N
56 b   G
57 b   V
58 b   I    S      86
59     T +  D      87
60 a   K -  I      88
61 a   D +  T a    89
62 a   E £  A a    90
63 a   A *  S a    91
64 a   E *  V a    92
65 a   K *  N a    93
66 a   L £  C a    94
67 a   F @  A a    95
68 a   N *  K a    96
69 a   Q *  K      97
70 a   D @  I      98
71 a   V £  V      99
72 a   D *  S     100
73 a   A +  D     101
74 a   A
75 a   V
76 a   R :  G     102
77 a   G    D     103
78 a   I
79 a   L
80 a   R
81     N
82 a   A
83 a   K
84 a   L
85 a   K
86 a   P
87 a   V
88 a   Y
89 a   D
90 a   S
91     L
92     D
93 a   A
94 a   V
95 a   R
96 a   R
97 a   C
98 a   A
99 a   L
```

```
 LZM          LYZ

100 a I
101 a N
102 a M
103 a V
104 a F
105 a Q
106 a M
107   G
108 a E . G    104
109 a T
110 a G
111 a V
112 a A
113 a G
114   F
115 a T
116 a N
117 a S
118 a L
119 a R
120 a M
121 a L
122 a Q
123 a Q
124   K
125   R
126 a W
127 a D
128 a E
129 a A
130 a A
131 a V
132 a N
133 a L - M    105
134 a A : N    106
135   K : A    107
136   S - W    108
137 a R - V    109
138 a W - A    110
139 a Y - W    111
140 a N - R    112
141 a Q = N    113
142   T - R    114
143 a P
144 a N
145 a R
146 a A
147 a K
148 a R
149 a V + C    115
150 a I - K    116
151 a T   G    117
152 a T : T    118
153 a F
154 a R
155 a T . D    119
156   G   V    120
157   T . Q    121
158   W : A    122
159   D
160   A . W    123
161   Y : I    124
162   K : R    125
163   N   G    126
164   L   C    127
          R    128
          L    129
```

(a)

Figure 7. (a) Alignment of the phage T4 lysozyme with hen egg-white lysozyme. The annotation of the alignment is described in the legend to Fig. 3. Hen egg-white lysozyme is denoted by LYZ and the phage protein by LZM. The alignment of the β-sheet region (residues 38 to 61 in LYZ, 13 to 34 in LZM) differs from that of Rossman & Argos (1976) and Weaver *et al.* (1985) (see the text) and is examined structurally in (b).



(b) Superposition of the β-sheets from hen egg-white and phage T4 lysozymes. The β-sheet regions defined in (a) were superposed interactively using QUANTA. These are represented using virtual C^{α} bonds that connect filled atoms in the hen structure and open circles in the phage structure. Some equivalent atoms are joined by broken lines. The residues discussed are indicated by their 1-letter amino acid codes: these include Ile55 in the hen structure, which is inserted in the second β-turn, and Leu33 in the phage structure, which, by the definitions of secondary structure by the method of Kabsch & Sander (1983), forms a β-bulge (c). This bulge was not found in the alignment in (a).

(c) Hydrogen-bonding map of lysozyme β-sheets. Hydrogen bonds as defined by the method of Kabsch & Sander (1983) are shown (as broken lines) for the β-sheets of (1) hen egg-white lysozyme (LYZ) and (2) phage T4 lysozyme (LZM). The sheets correspond to the regions superimposed in (b) and equivalent atoms in the 2 structures occupy equivalent positions in the diagrams. Ile55 in LZM, which has no corresponding position in LYZ, is circled.

Fig. 7.

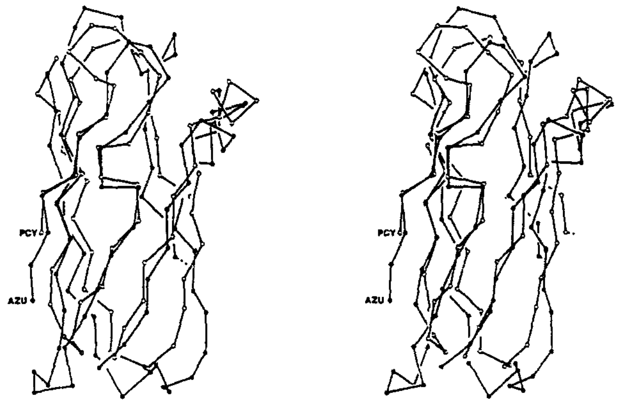
determined with considerable difficulty, often requiring extensive use of interactive computer graphics facilities to determine and verify the reported correspondence. To repeat this process automatically using a program that requires relatively little computer time is, we believe, a significant advance in the methodology of structure comparison.

An important aspect of our method that distinguishes it from previous approaches is its insensitivity to the displacement of equivalent substructures between the molecules being compared. This advantageous behaviour arises because the comparison of atoms that are spatially adjacent to any pair of residues being compared will score more highly than remote atoms, thus giving greater prominence to local structure. Through this effect, altering the contrast between high and low scores can effectively introduce a distance-dependent bias into the score calculation. This could either reduce the contribution of long-range effects, forcing the method to attach greater importance to local structure, or could be reversed, putting greater emphasis on global structure at the expense of local conformation. The effect was used above to improve the alignment of the remotely related structures of the immunoglobulin constant and variable domains, and a more systematic evaluation of its potential is in progress.

The ability to find the proper equivalence of displaced substructures is important in the characterization of regular supersecondary motifs. For example consider two β-hairpins each with, say, a type-I turn, but with one hairpin containing two extra residues in its β-strands. A conventional superposition approach might have difficulty in

	PCY			AZU		
				C	b	3
				S	b	4
1	b	I	£	V	b	5
2	b	D	*	D	b	6
3	b	V	£	I	b	7
4	b	L	*	Q	b	8
5	b	L	*	G		9
6	b	G				
7	b	A	=	N		10
8		D				
9		D				
10	b	G	:	D		11
11	b	S	=	Q		12
12	b	L	*	M		13
13	b	A	*	Q		14
14	b	F	@	F		15
15	b	V	+	N		16
16		P	+	T		17
17	b	S	@	N	b	18
18	b	E	*	A	b	19
19	b	F	=	I	b	20
20	b	S	=	T	b	21
21	b	I	=	V	b	22
22		S	-	D		23
23		P	:	K		24
				S		25
				C		26
				K	b	27
24		G	.	Q	b	28
25	b	E	-	F	b	29
26	b	K	=	T	b	30
27	b	I	+	V	b	31
28	b	V	£	N	b	32
29	b	F	*	L	b	33
30	b	K	£	S	b	34
31	b	N	+	H	b	35
32	b	N	=	P		36
33		A	.	G		37
				N		38
34		G	.	L		39
				P		40
				K		41
				N		42
				V		43
35		F	.	M		44
36	b	P	*	G		45
37	b	H	£	H		46
38	b	N	+	N		47
39	b	I	£	W		48
40	b	V	£	V		49
41	b	F	*	L		50
42	b	D	:	S		51
43	b	E				
44	b	D				
45	b	S				
46	b	I				
47	b	P		T		52
48		S		A		53
49		G		A		54
50		V	.	D		55
51		D				
52	a	A	.	M		56
53	a	S	.	Q		57
54	a	K	.	G		58
				V		59
				V		60
				T		61
				D		62
				G		63
				M		64
				A		65
				S		66
				G		67
				L		68
				D		69
				K		70
				D		71
				Y		72
				L		73
				K		74
				P		75
				D		76
				D		77
				S		78
				R		79
				V		80
				I		81
55	a	I	:	A		82
56	a	S	-	H		83
57		M	=	T		84
58		S				
59		E				
60		E				
61		D	=	K		85
62		L	+	L		86
63		L	*	I		87
64		N	:	G		88
65		A		S		89
66		K	.	G		90
67	b	G				
68	b	E	+	E	b	91
69	b	T	+	K	b	92
70	b	F	+	D	b	93
71	b	E	+	S	b	94
72	b	V	=	V	b	95
73	b	A	:	T	b	96
74	b	L	:	F	b	97
75		S	.	D	b	98
				V	b	99
				S		100
				K		101
				L		102
76		N	.	K		103
77		K	:	E		104
				G		105
78	b	G	.	E	b	106
79	b	E	:	Q	b	107
80	b	Y	=	Y	b	108
81	b	S	*	M	b	109
82	b	F	@	F	b	110
83	b	Y	£	F	b	111
84	b	C	@	C	b	112
85		S	*	T		113
86		P	=	F		114
				P		115
				G		116
87		H	*	H		117
88		Q	£	S		118
89		G	:	A		119
90		A				
91		G	-	L		120
92	b	M	*	M	b	121
93	b	V	£	K	b	122
94	b	G	@	G	b	123
95	b	K	@	T	b	124
96	b	V	+	L	b	125
97	b	T	+	T	b	126
98	b	V	+	L	b	127
99	b	N				

(a)



(b)

Figure 8. (a) Alignment of the copper-binding proteins plastocyanin and azurin. The symbols used are as in the legend to Fig. 3. β-Strands from sheets 1 and 2 (Lesk & Chothia, 1982) are labelled b and B, respectively. The register of the sheets is in agreement with that of Adman *et al.* (1985).

(b) Superposition of plastocyanin and azurin. Stereo diagrams showing the structures of plastocyanin (open α-carbons) and azurin (filled α-carbons). The proteins were superposed interactively using QUANTA to illustrate the alignment in (a). The 2 sheets are seen edge-on to emphasize the register between them. Some loops at the back of the molecules that have no close structural similarity (corresponding to the large insertion in azurin) have been removed for clarity. The N termini are marked by PCY and AZU for plastocyanin and azurin, respectively.

distinguishing between the two following alignments:

```
● · ● · ●○○● · ● · ●
    ● · ●○○● · ● ·
```

and:

```
● · ● · ●○○● · ● · ●
● · ●○        ○● · ●
```

where dots of similar size lie on the same side of the sheet and open circles are the turn residues. (The two possibilities are superpositions based on different registers of the β-sheet.) If the second alignment of the sheet residues is correct then the turn residues are not identified as equivalent. However, as our method is sensitive to the environment and orientation of each residue, it has the ability to match the turn residues correctly, as follows:

```
● · ● · ●○○● · ● · ●
● · ●     ○○     ● · ●
```

Our method also allows the simultaneous comparison of structure and sequence. Interestingly, we found this aspect to be less useful than anticipated. From our preliminary investigations, this appears to result from the general observation that structure is better conserved through evolution than is sequence. Thus, where the sequences are sufficiently close to be meaningfully compared, the structures are usually closer and therefore dominate the comparison. However, a situation where the sequence had a beneficial effect was seen in the comparison of the immunoglobulin variable and constant domains. The sequence bias helped to align the conserved tryptophan in the C strand but had little effect on the E–F hairpin. Indeed in the latter region the structural misalignment often produced a good sequence alignment. With these structures the correct alignment of the sheets can be determined from the positions of the disulphide cysteines, but the correct equivalence may not always be so clear and situations will, undoubtedly, be encountered where a good structural similarity conflicts with a good sequence similarity.

The treatment of gaps in sequence between equivalent structures is not a subject that has been thoroughly investigated here, since most of the problems considered above aligned well with the gap penalty that we initially assigned and seemed insensitive to variation in this. For more remotely related comparisons it may, however, be necessary to investigate both the size and form of the gap penalty in more detail. Following the work of Schulz (1977) on structure comparison, the possibility of allowing a residue in one structure to be equated with more than one in the other may also be a useful approach. The dynamic programming equivalent of this is referred to as "time warping" (Kruskal & Liberman, 1983), since the variable axis is normally time rather than a biological sequence. Its implementation would require only trivial changes from our current formulation.

As in the field of sequence comparison, an important aspect of our future work must be to develop statistical criteria for evaluating the significance of structural comparisons. This will involve the comparison of large numbers of structures, which could be generated automatically (Cohen & Sternberg, 1980; Schulz, 1980) or might entail the pairwise comparison of all known structures. Such a study would define a structural relationship between proteins, allowing families to be rigorously defined. Following the lead of Richards & Kundrot (1988), common motifs might then be extracted in an idealized form to produce a more symbolic classification. As our method is insensitive to insertions and deletions, the size of motifs could be large, including structures such as the Rossman fold, Greek key or β/α-barrel.

Although we concentrated our analysis of structure at the residue level, this is not an inherent restriction in the method. Any level of structural organization could be compared, ranging from an all-atom representation (which may be useful for comparing side-chain packing) to a more abstract representation of secondary structures that could be useful in comparing chain folds. Perhaps most importantly, our method has formulated the structure comparison problem in a methodology that is common to that used for the comparison of protein sequences. This allows the sophisticated methods developed for sequence comparison, including fast fragment-based techniques and multiple comparison methods, to be applied to protein structure alignment.

In conclusion, we have taken an important numerical method that had hitherto been used to compare only one-dimensional objects (sequences) and generalized it for the comparison of structures of any dimensionality, specifically three-dimensional protein structures. However, the method can be applied to any spatial pattern-matching problem that contains a linear constraint, and we expect it to find many applications in both the area of protein structure analysis and beyond.

We thank Drs N. Michael Green and Michael S. Waterman for valuable comments on the manuscript.

References

Adman, E. T. (1985). In *Metalloproteins* (Harrison, P. M., ed.), part I, chap. I, pp. 1–142. Verlag Chemie, Weinheim.

Adman, E. T. & Jensen, L. H. (1981). *Israel J. Chem.* **21**, 8–12.

Amzel, L. M. & Poljak, R. J. (1979). *Annu. Rev. Biochem.* **48**, 961–997.

Argos, P. (1987). *J. Mol. Biol.* **193**, 385–396.

Bernstein, F. C., Koetzle, T. F., Williams, G. J. D., Meyer, E. F., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanochi, T. & Tasumi, M. (1977). *J. Mol. Biol.* **112**, 535–542.

Blake, C. C. F., Koenig, D. F., Mair, G. A., North, A. C. T., Phillips, D. C. & Sarma, V. R. (1965). *Nature (London)*, **206**, 757–761.

Chothia, C. & Lesk, A. M. (1982). *J. Mol. Biol.* **160**, 309–323.

Chothia, C. & Lesk, A. M. (1986). *EMBO J.* **5**, 823-826.

Cohen, F. E. & Sternberg, M. J. E. (1980). *J. Mol. Biol* **138**, 321–333.

Dayhoff, M. O. (1978). *Atlas of Protein Sequence and Structure.* suppl. 3, National Biomedical Research Foundation, Washington, DC.

Gariepy, Y. & Hodges, R. S. (1983). *FEBS Letters*, **160**, 1–5.

Garrett. T. P. J., Clingeleffer, D. J., Guss, J. M., Rogers, S. J. & Freeman, M. C. (1984). *J. Biol. Chem.* **259**, 2822–2825.

Hubbard, T. J. P. & Blundell, T. L. (1987). *Protein Engineering*, **1**, 159–171

Kabsch, W. & Sander, C. (1983). *Biopolymers*, **22**, 2577–2637.

Jones, T. A. & Thirup, S. (1986). *EMBO J.* **5**, 819–822.

Kretsinger, R. H. (1980) *Crit. Rev. Biochem.* **8**, 119–174.

Kruskal, J. B. & Liberman, M. (1983). In *Time Warps, String Edits and Macromolecules* (Sankof, D. & Kruskal, J. B., eds), chap. 4, pp. 125–161, Addison-Wesley, Reading, MA.

Kuntz, I. (1975). *J. Amer. Chem. Soc.* **97**, 4362–4365.

Lesk, A. M. & Chothia, C. (1980). *J. Mol. Biol.* **136**, 225–270.

Lesk, A. M. & Chothia, C. (1982). *J. Mol. Biol.* **160**, 325–342.

Matthews, B. W. & Rossman, M. G. (1985). *Methods Enzymol.* **115**, 397–420.

Matthews, B. W., Remington, S. J., Grutter, M. G. & Anderson, W. F. (1981). *J. Mol. Biol.* **147**, 545–558.

Moews, P. C. & Kretsinger, R. H. (1975). *J. Mol. Biol.* **91**, 201–228.

Needleman, S. B. & Wunsch, C. D. (1970). *J. Mol. Biol.* **48**, 443–453.

Nishikawa, K. & Ooi, T. (1974). *J. Theoret. Biol.* **43**, 351–374.

Padlan, E. A. & Davies, D. R. (1975). *Proc. Nat. Acad. Sci., U.S.A.* **72**, 819–823.

Phillips, D. C. (1970). *Biochem. Soc. Symp.* **31**, 11–28.

Ploegman, J. H., Drent, G., Kalk, K. H. & Hol, W. G. J. (1978). *J. Mol. Biol.* **123**, 557–594.

Rao, S. T. & Rossman, M. G. (1973). *J. Mol. Biol.* **76**, 241–256.

Remington, S. J. & Matthews, B. W. (1978). *Proc. Nat. Acad. Sci., U.S.A.* **75**, 2180–2184.

Remington, S. J. & Matthews, B. W. (1980). *J. Mol. Biol.* **140**, 77–99.

Remington, S. J., Ten-Eyck, L. F. & Matthews, B. W. (1977). *Biochem. Biophys. Res. Commun.* **75**, 265–270.

Richards, F. M. & Kundrot, C. E. (1988). *Proteins*, **3**, 71–84.

Rossman, M. G. & Argos, P. (1975). *J. Biol. Chem.* **250**, 7525–7532.

Rossman, M. G. & Argos, P. (1976). *J. Mol. Biol.* **105**, 75–96.

Rossman, M. G. & Argos, P. (1977). *J. Mol. Biol.* **109**, 99–129.

Sankof, D. & Kruskal, J. B. (1983). Editors of *Time Warps, String Edits and Macromolecules*, Addison-Wesley, Reading, MA.

Schulz, G. E. (1977). *J. Mol. Evol.* **9**, 339–342.

Schulz, G. E. (1980). *J. Mol. Biol.* **138**, 335–347.

Szebenyi, D. M. E. & Moffat, K. (1985). *Nature (London)*, **294**, 327–332.

Taylor, W. R. (1986). *J. Mol. Biol.* **188**, 233–258.

Taylor, W. R. (1988). *J. Mol. Evol.* **28**, 161–169.

Taylor W. R. & Orengo, C. A. (1989). *Protein Engineering*, in the press.

Waterman, M. S. (1984). *J. Theoret. Biol.* **108**, 333–337.

Weaver, L. H., Grutter, M. G., Remington, S. J., Gray, T. M., Issacs, N. W. & Matthews, B. W. (1985). *J. Mol. Evol.* **21**, 97–111.

Edited by R. Huber