Protein Engineering vol.12 no.3 pp.203–216, 1999

Protein structural domain identification

William R.Taylor

Division of Mathematical Biology, National Institute for Medical Research, The Ridgeway, Mill Hill, London NW7 1AA, UK

A simple method for the definition of protein structural domains is described that requires only α-carbon coordinate data. The basic method, which encodes no specific aspects of protein structure, captures the essence of most domains but does not give high enough priority to the integrity of β-sheet structure. This aspect was encouraged both by a bias toward attaining intact β-sheets and also as an acceptance condition on the final result. The method has only one variable parameter, reflecting the granularity level of the domains, and an attempt was made to set this level automatically for each protein based on the best agreement attained between the domains predicted on the native structure and a set of smoothed coordinates. While not perfect, this feature allowed some tightly packed domains to be separated that would have remained undivided had the best fixed granularity level been used. The quality of the results was high and, when compared with a large collection of accepted domain definitions, only a few could be said to be clearly incorrect. The simplicity of the method allowed its easy extension to the simultaneous definition of domains across related structures in a way that does not involve loss of detail through averaging the structures. This was found to be a useful approach to reconciling differences among structural family members. The method is fast, taking less than 1 s per 100 residues for medium-sized proteins.

Keywords: Ising model/protein structure domains

Introduction

It has been clear since the determination of the earliest protein structures (Phillips, 1966) that there is a level of structural organization that is greater than the folding of the chain into simple secondary structure components. The exact definition of these structural domains, however, has remained problematic since there is a continual progression from proteins that slightly divide into two lobes to those that form clearly distinct folded regions separated by a flexible linking segment of chain. A component of (sequentially) local organization is partly an element in the idea of a domain but is not sufficient as some domains are formed from segments of chain that are distant in the chain. Secondary structure, in particular the β-sheet, also influences the definition of a domain since β-sheets are rarely split into separate domains. However, although one sheet would not normally be in two domains, two or more sheets may be in one domain, so again, this structural feature does not provide a sufficient definition. A concept sometimes taken as a rough working definition of a domain is that, if excised, the domain should remain folded as a stable structure. Although difficult to test (either experimentally or computationally), an implication of this concept is that the excised domain should contain a hydrophobic core and should therefore be larger than, roughly, 40 residues (for reviews, see Janin and Wodak, 1983; Janin and Chothia, 1985).

All the above principles have, in various combinations, been taken to form operational definitions of domains. Local compactness was taken as the principle aspect in the early method of Rose (1979) and more recently has been extended by Holm and Sander (1994) in a way that captures the relationship between compact units. Swindells (1995a) concentrated more on the requirement of having a hydrophobic core in each domain (Swindells, 1995b), extending cores outwards from their deepest components and, where necessary, pruning and fusing these into larger units. Some older methods such as that of Rose (1979) and the more recent method of Siddiqui and Barton (1995) have focused on minimizing the number of chain breaks needed to separate domains while also measuring the degree of association between the separating units, while Rashin (1985) and Islam *et al*. (1995) employed solvent area calculations. Sowdhamini *et al*. (1996) also captured many of these ideas but at the level of secondary structure elements. Whatever their primary guiding principle, most of these methods apply corrections to their initial definitions on the basis of the remaining (secondary) principles. Typically, the primary method generates alternative definitions that can be selected using the secondary principles, which, for example, may involving counting the resulting breaks in the chain and secondary structures. Often, these secondary filters become a complex weighted combination (as, for example, in the method of Siddiqui and Barton, 1995).

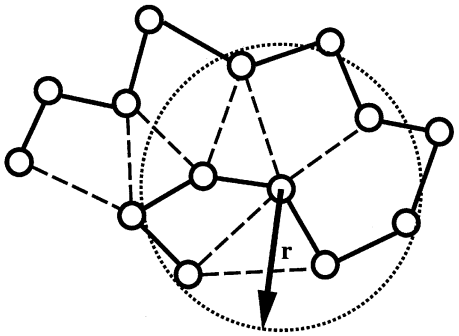
The methods described above generally take the approach in which a predefined idea of a domain is imposed on the structural data. In the language of systems analysis, this would be called a ‘top-down’ approach and the inherent danger in its application is being unable to recognize when the data do not fit the conceptual model. An alternative approach is to reverse the direction and let the idea emerge from the data, in what is sometimes called a ‘data-driven’ or ‘bottom-up’ approach. In this paper, a ‘bottom-up’ method for structure domain definition is described that is based on a very simple idea that has few parameters, so allowing their effect to be systematically investigated and, perhaps most important, it can be easily extended to the simultaneous definition of domains across homologous structures.

Methods

Ising model

The basic method is similar to an Ising model in which the structural elements of the model change state according to a function of the state of their neighbours. Although Ising models are typically applied to two states on a two-dimensional lattice, the approach has also been applied to the one-dimensional protein chain in the Zimm and Bragg (1959) model of helix–coil transitions (see Thouless, 1992, and Bruce and Wallace, 1992, for reviews of the approach applied to magnetic and more general phenomena, respectively). In the current method,

a



b

0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2	3	3	5	5	7	7	8	10	9	10	11	12
2	3	4	4	6	6	6	8	9	9	9	9	10	11
3	4	5	5	7	6	7	7	8	10	8	8	9	10
4	5	6	6	6	7	8	8	8	9	9	9	8	9
5	6	7	6	7	7	7	8	9	8	8	8	7	8
6	6	6	7	8	7	8	7	8	9	7	7	8	7
7	7	7	6	7	8	7	8	7	8	8	8	7	8
7	6	7	7	8	7	8	7	8	7	7	7	8	7
7	7	8	7	7	8	7	8	7	8	8	8	7	8
8	8	7	8	8	7	8	7	8	7	7	7	8	7

Fig. 1. State-label evolution in a small domain. (**a**) A schematic protein backbone is shown as connected α-carbons (full lines). Neighbouring residues are defined to lie within a radius *r*. In this simplified example, neighbours are indicated by dashed lines (connections between *i* and *i* + 2 have been omitted for clarity). (**b**) Starting with consecutive state labels (top line), these are modified through successive cycles (following lines) as described in Equations 1 and 2 (in this simple example, however, inverse distance weighting is not applied so the process can be followed more easily 'by hand'). The final state oscillates between 7 and 8 and, in the full method, an average is taken over consecutive cycles to attain a steady state.

the neighbours are defined by spatial proximity in the three-dimensional protein structure, and the states are multi-valued labels. In this implementation, the approach has affinity to the analysis of protein structure using connection topology (Aszodi and Taylor, 1993).

Basic method. Each residue in the protein chain is assigned a numeric label. If a residue is surrounded by neighbours with, on average, a higher label, then its label increases, otherwise it decreases. This test and reassignment are applied repeatedly to each residue in the chain. A worked example is shown in Figure 1.

Representing the sequences of labels as $S = \{s_1, s_2, \ldots, s_N\}$, for a protein of length N, then the iteration can be stated as

$$s_i^{t+1} = s_i^t + \mathcal{U}\left[\sum_{j=1}^{N} \mathcal{J}(s_i^t, s_j^t)\right], \forall i = 1 \ldots N \qquad (1)$$

At each iteration t, the new state of residue i (at $t + 1$) is determined by the influence of all other residues (j) modified by the function $\mathcal{J}$ which is referred to as the coupling function. Where the function is simple multiplication, then the state revision can be represented by a matrix multiplication as in the Zimm–Bragg method (Zimm and Bragg, 1959). The function $\mathcal{U}$ takes the sum over the neighbours and transforms it to either +1 or –1 for positive and negative sums, respectively.

Coupling function. The coupling function ($\mathcal{J}$ in Equation 1) calculates the inverse distance between the α-carbons of residues i and j and returns a negative value if the state label of i (s_i) is less than that of j (s_j). An upper limit (radius, r) on the proximity of the neighbours was imposed on those taken into the calculation. Explicitly, the function evaluates the expression to which it is equivalenced below:

$$\mathcal{J}(s_i, s_j) \equiv \begin{cases} p_{ij} \text{ if } s_j > s_i \text{ and } d_{ij} < r \\ -p_{ij} \text{ if } s_j < s_i \text{ and } d_{ij} < r \\ 0 \text{ otherwise} \end{cases} \qquad (2)$$

where d_{ij} is the interatomic distance between the α-carbons of residues i and j, p_{ij} is the inverted distance r/d_{ij} and r is the neighbourhood radius. The inverted distances constitute a matrix (**P**) which will be of further use below.

Some trials were made with different functions, in particular, with no inverse weighting (giving a simple majority 'vote' in Equation 1) and with inverse-squared weighting. The results for both were remarkably similar to the basic method but the latter appeared to undervalue the contribution of neighbours whereas the former increased the sensitivity of the result of the choice of cutoff radius (r). This behaviour is typical of Ising models in which the details of the lattice and the form of the coupling function make little difference to the global properties (Bruce and Wallace, 1992). The choice of the cutoff radius will be considered below but its use as a scaling factor (in Equation 2) does not affect the result since only the final sign of the sum is considered in Equation 1.

Label assignment. The most obvious choice for label assignment is the sequential residue number itself. This naturally embodies the desired property that sequentially adjacent residues will be predisposed to belong to the same domain. Other schemes will be considered below but, unless stated otherwise, simple residue numbering should be assumed.

Model evolution and domain extraction

The recursive iteration of Equation 1 results in compact regions evolving towards the same residue number. However, if there are two compact regions linked by a long exposed segment of chain (to take an extreme example), then each domain will evolve towards a local value and these labels will extend and meet half way along the linker. At this point, neither side will have sufficient 'leverage' to 'convince' the other to adopt its label and the system will cease to evolve (typically oscillating, or 'flickering' at the point of label discontinuity). For the extraction of domain definitions it is necessary that this stage in the evolution is detected and the iteration terminated, allowing the assignment of residues with a common label as a domain. To do this, some minor technical difficulties need to be addressed.

Stopping the iteration. Because of the potential for domain boundaries to 'flicker', the iteration cannot simply be terminated when there is no further change in labelling. This problem was overcome by keeping an average over two cycles and monitoring the squared deviation of this between successive cycles (summed over the length of the sequence). Any simple oscillation will thus be averaged out; for example, if a residue position alternates between 8 and 7 on successive cycles, then the difference in successive averages (8 + 7, 7 + 8, ...) will be zero. The iteration was stopped when the mean squared deviation of the average between successive cycles was less than 10^{-6} or if this degree of convergence was not obtained, then the calculation was stopped after a number of iterations equal to half the number of residues in the protein. This gives sufficient opportunity for both the amino and carboxy termini to evolve to a common label if they lie in the same domain.

Refining unique labels. In the basic method, the labels evolve

in discrete unit steps. This admits the small possibility that two independent domains might converge to the same value by chance. This possibility can be minimized by using a smaller step (increment/decrement) size but, if a small step size is selected at the outset, then the evolution of the system will be very slow. A compromise was made by following the initial evolution of the system by a further set of $N/2$ iterations (where N is the number of residues in the protein) in which the step size decreased linearly from one to zero. After this, the value of the labels within a domain generally agreed to better than 10^{-2}, greatly reducing the chance of two domains having the same label within the error of convergence.

Conforming to expectation

The method as described to this point, when applied to a variety of representative proteins performed remarkably well, especially considering that it embodies no encoding of any feature specific to proteins (details of these results will be provided below). However, as discussed in the Introduction, there are some assumptions in the received definition of a domain that need to be taken into account to produce a definition that conforms to expectation. Principal among these is the expectation that (1) the chain should not pass too frequently between domains, (2) small domains should be ignored or avoided and (3) secondary structure, in particular β-sheets, should not be broken.

Reclaiming short loops. Examination of some of the initial test results revealed that most of the frequent chain crossings between domains resulted simply from short loops 'dipping' in and out of the adjacent domain. These could easily be 'corrected' by resetting their label to that of the flanking domain; however, situations can be imagined where it is not obvious which loop should be reset, as illustrated by the following example in which two domains (with labels 5 and 7) mingle: {...5555577755577777...}. Simple smoothing (taking an average over a window) cannot be used as this would alter the residue labels; however, a solution was found by using a form of smoothing based on the median, rather than the mean, in which the position in the centre of a window is replaced by the median of the values in the window (Bangham, 1988). This method, when iterated to completeness, levels all peaks (or troughs) less than half the window size, but these are flattened (or filled) only with observed values so no new domain labels are created by the process. A window size of 21 was taken, eliminating all excursions of 10 or less residues.

Reassigning small domains. As in other studies (Siddiqui and Barton, 1995; Jones *et al*., 1998), domains of $\leqslant 40$ residues were not accepted. These might simply be ignored (marked as unassigned regions), but it was considered better to see if they might be joined on to another existing domain. This was done using a variant of the core calculation in Equation 1 in which each residue in the small domain was directly assigned the (weighted) mean values of its neighbours, as follows:

$$s_i^{t+1} = \sum_{j=1}^{N} (s_j^t p_{ij}) / \sum_{j=1}^{N} p_{ij}, \forall s_i^t < 0 \qquad (3)$$

where p_{ij} is an element from the matrix of reciprocal distances **P**. Reassignment was made only for residues that shared a common label with <40 others and this was 'flagged' by setting the label of all such residues to –1 (hence the condition $\forall s_i^t < 0$). Although not explicitly stated above, as before, the sum was taken only over residue pairs closer than the cutoff radius r and, in addition, residues in the process of being reassigned ($s_j^t < 0$) were also excluded.

After reallocation of small domains, the balance between the larger domains might have altered. This potential disequilibrium was allowed for by taking the new set of labels (S^{t+1}, calculated by Equation 3) as the starting point for another complete domain assignment calculation and the whole exercise was repeated until no small domains remained or to a limit of five times. This limit was sometimes reached as some small domains are truly isolated and remain 'unclaimed' by any of their larger neighbours. In this situation, it was considered unnecessary to introduce any further steps to 'force' their reallocation. Any remaining small domains were not included in the counts of domain numbers discussed below.

Keeping β-sheets intact. The basic method deviated most seriously from expectation in a propensity to divide large proteins that were dominated by a single β-sheet. This tendency was most apparent in the 8-fold alternating β/α-barrel proteins, which often have weakly packed strands and helices as a result of the stagger in hydrogen bonding around the barrel. Solutions to this problem have been found previously through the use of the recorded secondary structure information (extracted from the protein structure databank) or based on calculated hydrogen bonding. In the current method, a self-contained solution was sought that depended (as does the basic method) on the use of α-carbon coordinates alone.

For each residue i in the protein, its nearest and second nearest neighbours (j and k) were found, such that $d_{ij} < h$, $d_{ik} < h$, $|i - j| > 3$ and $|j - k| > 5$ (where d is an interatomic distance). When all these conditions are met, the three residues potentially align in a β-sheet as j–i–k. The same conditions, with the exception of the last, were then reapplied to the two sequentially adjacent triplets $i \pm 1$, $j \pm 1$ and $k \pm 1$, for which the signs were adjusted to minimize the interatomic distances. The resulting set of six residues thus lie in the expected arrangement of a β-sheet and this was recorded in a matrix of pairwise links (**B**, initially zero) by adding 1 to each of the pairs across the sheet (B_{ij}, B_{ik}, $B_{i \pm 1, j \pm 1}$, $B_{i \pm 1, k \pm 1}$) and along the strands ($B_{i,i \pm 1}$, $B_{j,j \pm 1}$, $B_{k,k \pm 1}$), choosing signs as above. After processing each residue in this way, the strongest pairwise links in **B** have a maximum value of 6 for residues that lie in centre of a large sheet, dropping to 1 for corner pairs. The cutoff distance h was chosen as 7.5 Å, being a point midway between the separation of hydrogen-bonded β-strands and the separation of stacked β-sheets (typically 5 and 10 Å, respectively).

A bias was given to maintain the integrity of β-sheets by setting the initial label of their component residues to a common value. For consistency, this was initially done using the basic method itself, by substituting the matrix **B** for **P** (in Equation 1). However, it was found that this approach also was still prone to split weakly linked sheets into domains so the variation employed to reassign small domains was used instead, in which each residue takes the weighted mean label of its neighbours, again, substituting the matrix **B** for **P** (in Equation 3):

$$s_i^{t+1} = \sum_{j=1}^{N} (s_j^t b_{ij}) / \sum_{j=1}^{N} b_{ij}, \forall i = 1, ..., N \qquad (4)$$

Unlike the reassignment of small domains, Equation 4 was iterated to convergence using the stopping criteria employed in the basic method (see Methods). No neighbour cutoff was

applied as this is already inherently encoded in matrix **B** and Equation 4 was evaluated only for linked residues ($\Sigma_{j=1}^{N} b_{ij} > 0$). This approach has the desired property that the entire network of linked residues is still not forced to adopt the same label and weakly (possibly spuriously) linked sheets can still remain distinct. It should also be noted that this procedure only provides a set of starting labels to which the basic method is applied (as described above) and this still has complete freedom to reassign any of the initial labelling.

It was also considered whether an equivalent bias should be applied to α-helices; however, long helices often pack against more than one domain and it seemed more natural that these should be allowed to partition freely as dictated by the basic method.

Setting the granularity level

The basic method has only one parameter which is the neighbourhood cutoff radius (r). The value of r affects the average size of the resulting domains (and can be associated with the correlation length in the application of Ising models to critical-point phenomena). When r is small the resulting domains tend to be smaller but the relationship is not direct and, even when r is infinite, clear domains will still remain separated. Almost all the methods discussed in the Introduction have parameters that affect the granularity of the result but none have any mechanism for objectively setting this property, other than to optimize the parameters to give a result that approximates the definitions recorded in the literature. These, of course, will vary from author to author and, despite some attempts at homogenization, remain a heterogeneous standard.

One approach to this fundamental problem is to obtain two different (ideally independent) views and, when these agree, it can be assumed that some 'truth' has been found that is independent of any particular method. An approach along these lines was made by Jones *et al*. (1998) using three methods of domain identification. Unfortunately, it was found that, except for the most obvious examples these were never in full agreement (to better than 85% of equivalently assigned residues). An alternative approach is to use a single method but applied to homologous proteins. However, this is limited by the availability of homologues with sufficient structural difference to provide independent solutions. To circumvent this problem, a 'fake homologue' was created for each protein and the current method applied to both. This allowed the value of r to be varied and the level of granularity was accepted as the value where the two solutions agreed best.

Creating a 'fake homologue'. A simple way to create a structure with the same fold but differing in detail is to smooth the path of the chain. This technique has often been used to help visualize the fold of the chain, originally by Feldman (1976), and more recently (using the current algorithm) by Aszodi *et al*. (1995). Smoothing destroys almost all the specific details of protein geometry; however, for the current method this is not a disadvantage as it does not rely on any of these characteristic geometric features. Specifically, for each consecutive triplet of α-carbon coordinates, the central atom was replaced by the average coordinates of the triplet. This procedure was repeated five times giving a structure that was substantially different from the native coordinates but still recognizable. Although not directly comparable, the root-mean-square deviation between the smoothed and native chain was typically around 4–5 Å, which is equivalent to that found between analogous structures (having the same fold but no significant sequence similarity).

Table I. Domains without β-bias (see legend to Table III for details and summary)

			native structure							smooth structure							joint
protein	len	N	12	13	14	15	16	17	18	15	16	17	18	19	20	21	agree
1aak	150	1	[2]	2	2	2	1	1	1	[2]	1	1	1	1	1	1	95
1ace	526	1	[5]	3	2	4	4	3	2	[5]	2	3	2	2	2	2	95
1bbhA	131	1	1	1	1	[1]	1	1	1	2	2	2	[1]	1	1	1	
1bbpA	173	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1fxiA	96	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1gky	186	1	2	2	[3]	2	2	2	1	3	3	[3]	2	1	1	1	95
1gmfA	119	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1gmpA	96	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1gox	350	1	2	2	[2]	2	1	1	1	2	3	[2]	1	1	1	1	90
1ofv	169	1	1	2	[1]	1	1	1	1	2	2	[1]	1	1	1	1	
1pyp	280	1	3	2	[1]	1	1	1	1	2	1	[1]	1	1	1	1	
1rbp	174	1	2	2	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1rcb	129	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1rveA	244	1	2	1	2	[2]	1	1	1	3	2	1	[2]	2	1	1	94
1snc	135	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1tie	166	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1tlk	103	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1ula	289	1	2	2	[2]	2	1	1	1	2	2	[2]	2	2	1	1	95
1wsyA	248	1	2	[2]	2	3	1	1	1	2	[2]	2	3	1	1	1	93
2azaA	129	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
2ccyA	127	1	1	1	[1]	1	1	1	1	2	2	[1]	1	1	1	1	
2rn2	155	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
2stv	184	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
2tmvP	154	1	1	[1]	2	1	1	1	1	1	[1]	1	1	1	1	1	
3chy	128	1	2	2	[1]	1	1	1	1	2	2	[1]	1	1	1	1	
3cla	213	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
3dfr	162	1	1	2	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
4blmA	256	1	2	2	2	[1]	1	1	1	4	3	1	[1]	1	1	1	
5p21	166	1	[2]	1	1	1	1	1	1	[2]	2	2	2	1	1	1	96
1ezm	298	2	2	2	[2]	3	2	3	3	2	2	[2]	3	2	2	2	100
1lap	481	2	2	2	[2]	2	2	2	2	5	2	[2]	2	2	2	2	100
1pfkA	320	2	3	2	2	[2]	2	2	2	3	3	3	3	[2]	2	2	99
1ppn	212	2	2	[2]	2	2	[2]	2	2	2	[2]	2	2	2	2	2	98
1rhd	293	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	100
1sgt	223	2	2	2	[2]	1	1	1	1	2	2	[2]	2	2	1	1	96
1vsgA	362	2	4	2	[2]	2	2	2	2	3	2	[2]	3	2	2	2	98
1wsyB	385	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	93
2cyp	293	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	100
2gbp	309	2	3	2	[2]	2	3	3	2	3	2	[2]	2	2	2	2	96
2had	310	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	NO
3cd4	178	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	92
3gapA	208	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	99
3pgk	415	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	100
4gcr	174	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	99
5fbpA	313	2	3	3	3	3	1	2	2	3	3	3	2	2	2	2	98*
8adh	374	2	2	2	3	2	[2]	2	3	3	3	2	3	[2]	2	2	99
8atcA	310	2	3	2	2	2	2	2	2	3	3	3	3	3	3	3	NO
8atcB	146	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	100
1phh‡	394	2	2	2	2	[2]	2	2	2	2	4	4	3	[2]	2	2	99
3grs	461	3	4	3	[3]	3	3	3	3	3	3	[3]	3	2	2	2	99
8acn	753	3	4	2	3	3	3	[3]	2	3	3	4	1	4	[2]	1	87
1atnA†	372	24	2	2	[2]	2	3	2	2	3	2	[2]	2	2	2	2	99
3pmgA	561	4	4	4	[4]	4	3	3	3	4	5	[4]	4	4	4	4	97

Comparing domain agreement. Comparing the domains assigned with the smooth and native chains, it was apparent that the smooth chains required a slightly larger cutoff radius to give roughly comparable results. This compensates for the reduced interatomic contact in the smooth chain, especially in regions of α-helix packing where the helices have been reduced to almost straight lines. Values of $r + 3$, $r + 5$ and $r + 7$ were tested and a bonus of 3 was found to be sufficient.

Following Jones *et al*. (1998), a matrix of common residue counts in all domain pairings was compiled. The best overall count was then extracted from this matrix; however, where Jones *et al*. (1998) appear to assume that these values lie on the diagonal, a more general (but still 'greedy') algorithm was used in the current work, which has previously been described in the alignment of multiple sequences (Taylor, 1987). Since the smooth and native structures have the same number of residues, the number in agreement was reported as the percentage of the length of the protein.

Finding the best granularity level. The simplest algorithm to find the best agreement between the two structures is to vary the cutoff radius and monitor the percentage domain agreement. However, this is computationally expensive and a more

Table II. Domains with β-bias (see legend to Table III for details and summary)

			native structure							smooth structure							joint
protein	len	N	12	13	14	15	16	17	18	15	16	17	18	19	20	21	agree
1aak	150	1	2	2	2	2	[1]	1	1	2	1	1	1	[1]	1	1	
1ace	526	1	3	4	4	4	4	2	1	3	2	3	2	2	2	2	NO
1bbhA	131	1	1	1	1	[1]	1	1	1	2	2	2	[1]	1	1	1	
1bbpA	173	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1fxiA	96	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1gky	186	1	2	3	2	2	[1]	1	1	3	3	2	2	[1]	1	1	
1gmfA	119	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1gmpA	96	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1gox	350	1	2	1	[1]	1	1	1	1	2	2	[1]	1	1	1	1	
1ofv	169	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1pyp	280	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1rbp	174	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1rcb	129	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1rveA	244	1	2	1	2	[2]	1	1	1	3	3	1	[2]	2	1	1	92
1snc	135	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1tie	166	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1tlk	103	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1ula	289	1	2	2	1	[1]	1	1	1	2	2	2	[1]	1	1	1	
1wsyA	248	1	2	3	1	[1]	1	1	1	2	2	2	[1]	1	1	1	
2azaA	129	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
2ccyA	127	1	1	1	[1]	1	1	1	1	2	2	[1]	1	1	1	1	
2rn2	155	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
2stv	184	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
2tmvP	154	1	2	2	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
3chy	128	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
3cla	213	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
3dfr	162	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
4blmA	256	1	1	1	1	[1]	1	1	1	2	2	2	[1]	1	1	1	
5p21	166	1	1	1	[1]	1	1	1	1	1	1	[1]	1	1	1	1	
1ezm	298	2	2	2	[2]	3	2	3	3	2	2	[2]	3	2	2	2	100
1lap	481	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	100
1pfkA	320	2	2	2	[2]	2	2	2	2	3	3	[2]	2	2	2	2	99
1ppn	212	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	90
1rhd	293	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	100
1sgt	223	2	[2]	1	1	1	1	1	2	[2]	2	2	2	1	1	1	98
1vsgA	362	2	4	2	[2]	2	2	2	2	3	2	[2]	2	2	2	2	98
1wsyB	385	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	94
2cyp	293	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	100
2gbp	309	2	3	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	99
2had	310	2	3	2	2	2	2	[1]	1	2	2	1	1	1	[1]	1	
3cd4	178	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	98
3gapA	208	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	99
3pgk	415	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	99
4gcr	174	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	99
5fbpA	313	2	3	3	3	3	1	2	2	2	2	2	2	2	2	2	95[§]
8adh	374	2	3	[2]	3	3	2	2	2	2	[2]	2	2	2	2	2	99
8atcA	310	2	3	[3]	2	2	2	2	2	3	[3]	3	3	3	2	2	97
8atcB	146	2	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	100
1phh[‡]	394	2	2	3	2	4	3	[2]	2	2	3	2	3	3	[2]	2	99
3grs	461	3	3	3	[3]	3	3	3	3	3	4	[3]	3	2	3	2	95
8acn	753	3	2	[3]	3	2	2	2	2	3	[3]	2	2	2	3	1	99
1atnA[†]	372	24	2	2	[2]	2	2	2	2	2	2	[2]	2	2	2	2	99
3pmgA	561	4	4	4	[4]	4	4	3	3	4	4	[4]	4	4	4	4	97

restricted search strategy was adopted. From trial runs it was found that most solutions lay in the middle to lower part of the range $r = 10$–20 (see Results and Table I). A start point was taken as $r = 14$ and a search expanded with alternating lower and higher values, in unit steps, and terminating when r fell below 10. If at any point during the search the two domain assignments had 90% or more coincidence, then the search was halted and the current solution accepted. Otherwise the best agreement point was recorded and if at the end of the search this was 85% or better, then its solution was accepted. If during the search both structures were reported as single domains, and no other solution agreed to better than 85%, the structure was taken to be a single domain. Structures for which no solution was found (either as single or multiple domains) were marked as unassignable.

Excluding badly broken β-sheets. The search for the best granularity level provides an opportunity to check the integrity of the β-sheet structure, allowing this to be controlled without affecting the operation of the basic method. If a β-sheet was found to be broken by separating the domains, in either the native of the smooth structure, then that solution was not accepted and the search continued, as described above. Splits in β-sheets were measured by the summed value of the pairwise terms in the β-sheet network (as calculated in Methods) over all the residue pairs between the two domains. (Note that β-sheet network was calculated only once on the native structure and the splits based on the smooth structure were assessed on the same network.)

As the current (or any) calculation of β-sheet structure can never be completely reliable, some tolerance is desirable in the strictness with which the integrity of β-sheets is maintained. It was estimated that this should relate to the size of the protein, and a rough and reasonably generous level was set as the square root of the length of the protein. A small protein of 100 residues can therefore tolerate an error of 10 (it would cost 12 to split a 4×4 sheet in two), whereas a large protein of 400 can accept double this error.

Re-parsing domains

The domains defined by the basic method were automatically represented to the method to check if they could be further divided. This was done by linking the broken ends of the chain in the excised fragments and treating them as a new 'intact' protein. These reconnections were necessary, since a protein with chain breaks would behave differently in the Ising model from an equivalent connected chain (they also make the excised domains much easier to visualize). The connecting loops were 'grown' recursively from the broken ends in the direction of the centroid of the deleted segment until they came within bonding range. This was implemented by the following pseudo-code:

```
connect (A,B,C)
        put (A)
        if dist(A,B) > 5
                A = extend (A,C)
                B = extend (B,C)
                D = (A+B)/2
                if dist(A,B) < 3
                        put (D)
                else
                        C = (C+D)/2
                        connect (A,B,C)
                end if
        end if
        put (B)
end connect
```

The upper-case characters represent atomic coordinate vectors. The function add writes the atom position of its argument to the coordinate (PDB) file, 'extend' calculates a coordinate 3.8 Å beyond its first argument in the direction of its second and 'dist' calculates the Euclidean separation between its arguments. The starting coordinates (A, B and C) were (respectively) the first and last atoms in the deletion and the centroid of the deleted segment (excluding its first and last atoms).

Simultaneous definition on multiple structures

The flexibility of the labelling system allows the labels to be taken not simply as a residue position in a single structure but as a position in a multiple sequence alignment. Rather than suffer the distortions inherent in defining domains on an averaged multiple structure, or taking the averaged domain definitions after individual domain definition, the current method can allow simultaneous (interacting) domain definition across all the structures. This was achieved using the basic method on each structure individually but with the labels

derived from a multiple structure alignment. Between each iteration of Equation 1, the individually evolving labels can be synchronized by taking an average over the label values at each position across the structures that are represented at that point in the multiple alignment.

Structural data

Protein structural data were taken from the Protein Structure Databank (PDB) (Bernstein *et al*., 1977) as it existed in June 1998. This constituted 29 000 chains which were 'sifted' in successive runs of the multiple sequence alignment program MULTAL [using parameters described by Taylor (1998), Appendix II] until no chains remained with more than 50% residue identity when aligned. The selection criteria for those retained were based on a weighted combination of resolution, mean *B*-value (indicative of refinement) and their assorted properties, details of which can also be found in Taylor (1998) (Appendix II). After filtering, 1775 chains remained which will be referred to below as the PDB50 set.

Reference domain definitions were found in the collection of Islam and Sternberg at the ICRF (Imperial Cancer Research Fund) web-site: http://bonsai.lif.icnet.uk/domains/assign.html. The domain definitions in this collection have been extracted from the original literature, with some modification where necessary. A number of the files in this collection have been replaced in the current PDB and, rather than attempt to reconcile these, only those that had a current PDB file were used. Those remaining were then filtered, as above, to give a collection of 517 proteins in which no pair had better than 50% identity (referred to as the ICRF50 set).

For preliminary testing, a sub-set of the ICRF collection described by Jones *et al*. (1998) was taken. Only one of these had been replaced in the current PDB and for this the corresponding revised entry was taken. One member of this set, 1brd (bacteriorhodopsin), was rejected, not because it is an integral membrane protein, but because it has no loops connecting its transmembrane helices (this feature would hinder the evolution of the labels). This collection will be referred to as the UCL-subset.

Results

Tests on the UCL-subset of structures

Testing the basic method. The basic method was applied to the UCL-subset with different values of the neighbour cutoff radius *r* and the results were compared with those expected. This test set contains roughly half single domain proteins and multi-domain proteins and different behaviour was observed in each group.

On both smooth and native structures, both with and without the β-sheet bias, the single domain proteins were most correctly predicted at the higher values of neighbourhood radius, as would be expected. Values of $r = 18$ and $r = 20$ (or 21) for the native and smooth structure, respectively, gave perfect assignment with the exception of 1ace (acetylcholinesterase). However, this large protein (526 residues) looks like two domains but its obvious division splits a large β-sheet that runs through the structure.

By contrast, the multi-domain group showed little variation in error with different radii. Good predictions were made for the native structure at lower radii (13, 14) and at higher values with the smooth structure. The average assignment accuracy was 75% and better than 80% in the mid-ranges of *r* (see Tables I and II for details and Table III for a summary). A persistent deviant in these assignments was the protein 2had (haloalkane dehalogenase) which has a main β/α domain capped by an extended (bent) α-helix hairpin. This feature packs tightly on the β/α domain and by itself is not compact, making it an ambiguous candidate as a distinct domain.

Neglecting 1ace and 2had, the only error in the best assignment (smooth structure with $r = 20$) was 1sgt (elastin), which has two tightly packed β-barrels that have been accepted as domains for many years (McLachlan, 1979) but have previously been found difficult to split automatically into domains (Swindells, 1995a).

Automatic granularity adjustment. It was clear that tightly packed domains, such as those observed in elastin, require a lower level of granularity in their domain definition. If this is encouraged by reducing *r*, then the average accuracy of domain definition over the multi-domain proteins is largely unaffected but some of the single domain proteins begin to break up (see Table III). Most of these erroneous splits involve the division of a β-sheet and while this is discouraged (roughly twofold) by setting an initial β-bias, small proteins such as 1gky (guanylate kinase) which have two clear lobes (similar to adenylate kinase) re-establish a double domain split. However, such splits incur a high error score based on the summed value of the bonds broken in the β-network (the matrix **B** in Equation 4) and can be disallowed on this basis (see Methods), e.g. the

Table III. Summary of errors in Tables I and II

	native structure							smooth structure							joint
	12	13	14	15	16	17	18	15	16	17	18	19	20	21	
no β-bias															
single	12	12	9	7	2	2	1	14	12	7	6	3	1	1	7
multiple	7	2	2	3	4	4	5	7	6	5	8	4	5	5	$\frac{1}{2}$ (2)
total	19	14	11	10	6	6	6	21	18	12	14	7	6	6	$7\frac{1}{2}$ (2)
with β-bias															
single	8	6	4	4	1	1	0	10	9	6	3	2	1	1	1 (1)
multiple	7	3	3	6	4	5	5	3	4	3	5	6	2	4	1
total	15	9	7	10	5	6	5	13	13	9	8	8	3	5	2 (1)

Tables I and II show the number of domains calculated for the UCL sub-set of the ICRF domain collection on the native structure and the smoothed chain both without consideration of β-sheets (Table I) and with the bias to keep sheets intact (Table II) (see the main text for details). The PDB code is given for each structure (a terminal upper-case letter designates the chain) along with the number of residues in the protein (len) and the number of domains (N) a specified in the ICRF domain server (see the main text for location). ‡*P*-Hydroxybenzoate hydroxylase (1phh) differs in its number of domains in Jones *et al*. (1998) and the current ICRF server (the latter appears correct). †Actin (1atn) clearly has two domains (corresponding to an internal symmetry), each of which can be subdivided but the level of division is ambiguous (Holm and Sander, 1994; Siddiqui and Barton, 1995) and the protein has not been counted in this table. Without actin, the set contains 29 single domain proteins and 23 multi-domain proteins. The number of domains is shown for differing values of the neighbour cutoff radius *r* (Equation 2) ranging from 12 to 18 for the native structure and from 15 to 21 for the smoothed structure (equivalent values are 3 higher for the latter for reasons discussed in the main text). The point at which the domain definition best agree during the search strategy (Methods) is shown boxed for both structures. Where there is more than one domain, the percentage agreement (at the residue level) is shown under the heading joint agree. A NO in this column indicates that no agreement was obtained within the search range of 10–18. *Indicates that the solution was found at $r = 10$ and §at $r = 11$. The table summarizes the frequency of errors (on the basis of number of predicted and observed domains) for each structure, with and without the β-bias. The values in the joint column summarize the error over the boxed values in Tables I and II. This is only ambiguous for one protein (8acn), which has two or three predicted domains for the smooth and native structure, respectively. Since no structure has precedence, a value of $\frac{1}{2}$ was recorded. In this column, the number of times when no agreement was found is given in parentheses.

split of 1gky into two domains costs 26, twice as much as the permitted level.

Using the strategy outlined in Methods, a search for agreement between the native and smooth structures was started at $r = 14$, being just above the point at which the errors in the single-domain proteins begin to escalate. It was expected that spurious domain divisions would be less likely to agree at these lower levels of r and that this effect, combined with the tendency to avoid β-sheet disruption, would drive the single-domain proteins to find a solution at higher r values while still allowing the multi-domain proteins to find agreement at a lower granularity level.

With no β-sheet contribution, the results for the multi-domain proteins improved but remained much the same for the single-domain proteins. With the β-sheet bias and filter, the single domain proteins also improved to the same level and (neglecting 1ace) only one protein, 1rveA (ECO RV endonuclease) was incorrectly split. This protein has a carboxy-terminal α-helical extension that was split off from the main β/α domain. This also involved the removal of a distorted edge-strand from the β-sheet, but this β-strand has no links in the β-network matrix. Interestingly, no agreement was found for 1ace, which from the above discussion is an acceptable conclusion for this protein. Similarly, with the multi-domain proteins, 2had was predicted as one domain, despite having access to two-domain solutions at lower r values; however, these never agreed to the required level. Only one protein, 8atcA (aspartate transcarbamylase, A-chain), had the wrong number of domains. Investigation of the domain assignments revealed that, again, an edge β-strand had been split off along with some α-helices, giving three rather than the expected two domains. The correct two-domain solution existed at larger values of r and could be reached if the criterion for rejecting split sheets was stricter; however, this change was not implemented until a wider selection of proteins was investigated.

Application to the ICRF50 data set

The method was applied to the ICRF50 data set and full details of the results are reported in the Appendix. In summary, the errors found among these results include the division of two TIM barrels and the failure to split several multi-domain proteins. These included some representatives of the trypsin family, lactate dehydrogenase, a phosphate-binding protein (a periplasmic binding fold), an acid protease and a nitrogenase. Interestingly, all these proteins have relatives that were correctly divided.

Related 'errors' were found among the multi-domain proteins in which some of the larger proteins were only partially split into their accepted component domains but the remaining divisions could be obtained by reapplication of the algorithm to the fragments. This type of error brings to light a limitation in the search strategy employed to find agreement between the smooth and native chains. If there is sufficiently good agreement for a protein to be split into three domains, then the agreement at the level in which any two of these domains are combined cannot be worse and will therefore be accepted.

A solution to this problem is to perform a routine reapplication of the algorithm to each domain and assess the resulting additional splits. While this strategy would be beneficial to those included in Table X, about the same number of proteins again do not benefit from a reapplication, which leads to subdivisions at too fine a level (perhaps because the excised domains are not as compact as their equivalent native structures might be). However, a reasonable size limit can be imposed on the reapplication of the algorithm and if re-parsing is restricted to domains over 250 residues in length, then all those included in Table X can be accepted (1aozA, 1hkg, 1mioB, 1gsgP) with the exception that the symmetric thirds are not split in the N-terminal domain of 1eps. The only additional error introduced by this condition is the splitting of the TIM barrel in the α-amylase 2aaa.

Table IV. Multiple structure domain assignments

protein	PDB	len	RMSd	one	two	r	agree
pepsin	4pep	326	1.12 (321)	3	2/3	12	95%
cathepsin-D	1pplE	323	1.24 (311)	1	3	12	87%
cathepsin-D	1lybB	241	0.83 (234)	1	2	14	99%
renin	1mpp	357	1.20 (320)	NO	NO	-	3 best at $r = 15$ (81%)
rhizopuspepsin	2apr	325	1.17 (315)	NO	3	13	94%
retroviral	2hvc	203	1.75 (202)	2	2	14	99%

(a) pepsins (1rne)

protein	PDB	len	RMSd	one	two	r	agree
proteinase-A	2sga	181	1.30 (163)	1	1	14	2 at $r = 11$ (94%)
α-lytic protease	2alp	198	1.31 (168)	1	2	14	97%
protease-I	1arb	263	1.34 (201)	1	1	14	

(b) trypsins (1sgt)

protein	PDB	len	RMSd	one	two	r	agree
L-dehydrogenase	1lldA	313	1.01 (309)	1	1	14	
L-dehydrogenase	1ldnB	316	1.08 (313)	1	1	15	
M-dehydrogenase	1bdmA	317	1.60 (298)	1	1	16	2 at $r = 15$ (93%)†
M-dehydrogenase	1emd	312	1.54 (290)	1	1	14	

(c) dehydrogenase (6ldh)

Protein names with their PDB codes and length are followed by the number of domains defined on each single structure (one) and when combined with the proteins named below each sub-table (two). The weighted root-mean-square deviation (r.m.s.d.) is given for each structure pair (with the number of residues in parentheses). The percentage agreement between the native and smoothed structures is given under agree with the value of r at which the agreement was found. In (c), L = lactate and M = malate.

Multiple structure domain definitions

The outstanding problematic proteins all have structural relatives that have been correctly parsed into domains. Rather than search for a parameter combination that would satisfy all proteins, it was considered more sensible to use the redundant data found in these relatives to calculate a consensus domain definition. This was done as described in Methods, using a simple extension of the basic method, in which the undisrupted interactions in each protein were combined simultaneously.

Aspartyl protease family. The aspartyl protease family members (pepsins for short), which all have two domains, exhibit a wide variety of predicted domain structure, including one domain (1pp1E and 1lybB, both cathepsin-D), two domains (1rne, renin), three domains (4pep, pepsin) and two examples where a solution was not found (1mpp, renin; and 2apr, rhizopuspepsin). This family has previously been found to be difficult to parse (see, for example, Swindells, 1995a, for discussion).

The structure of renin 1rne, for which the correct double domain solution had been obtained, was combined with each of the others in turn and the results are given in Table IV. Much greater consistency can be seen among these results, but rather than converging towards the expected double-domain solution, most favour three domains and even where two domains were found (with 1lybB) there was a third domain

that was too small to declare. In all these results, the third domain formed an interface between the two pseudo-symmetric halves and is consistent with the analysis of domain movements by Sali *et al*. (1992).

Serine protease family. Although structurally not dissimilar to the pepsins, this family of smaller protease (referred to as trypsins, for short) exhibit the more consistent failure to split into the expected two domains. This behaviour is, again, typical of other automatic methods (Swindells, 1995a). The family includes proteinase-A (2sga), the alpha-lytic protease (2alp), achromobacter protease-I (1arb), trypsin (1sgt) and a virus coat protein (2tbv). Only the latter two were divided into two domains.

This family benefited little from the pair combinations reported in Table IV, with only 2alp joining 1sgt in a double domain solution.

Lactate dehydrogenase. Lactate dehydrogenase contains a tightly packed catalytic domain and a dinucleotide binding domain (DNBD), the latter having been identified since early times as a typical domain (Adams *et al*., 1970) because of its internal pseudo-symmetry and widespread recurrence in other nucleotide binding proteins (Rossman *et al*., 1974). The catalytic domain contributes a large carboxy-terminal helix to the DNBD and this interaction makes separation sufficiently difficult that all three lactate dehydrogenases considered (6ldh, 1lldA and 1ldnB) were predicted to be one domain. Indeed, if the granularity was reduced to try and separate the domains, the first half of the dinucleotide binding domain was the first part to be split off.

Combinations of the lactate dehydrogenases persisted in their single-domain solution. However, for their size, the 1 Å r.m.s.d. for the pairs does not constitute a large difference. More distantly related homologues were found in the malate dehydrogenases (1bdmA and 1emd), both of which had two domains correctly predicted but, even in combination with these, 6ldh maintained a single-domain solution. Examination of these results suggested that some of the failure to agree stemmed from the unique 20-residue unstructured N-terminal tail on 6ldh. This tended to slow the convergence of the first half of the Rossmann fold, increasing its likelihood of detaching as a separate domain. Removal of these residues led to one double domain solution with 1bdmA.

Other various pairs. Some less extensively related pairs of proteins were also encountered in the ICRF50 data set that gave different predicted domain definitions. These, along with any pair that also had the same but wrong prediction, were presented to the multiple structure algorithm.

Both versions of *Aspartate transcarbamylase* (8atcA and 2at2A) suffered the same loss of the edge part of one of their domains. Combining the two structures, this error was avoided as the best solution was found at a high or higher value of *r* than with the single structures.

The two chains of the molybdenum–iron nitrogenase (1mioA/B), which are distantly similar, had different predicted domain solutions. No solution was found for the A-chain while the B-chain was split into two followed by a further split. Combining the two chains immediately led to the correct split.

Application to the PDB50 data set

The algorithm was applied to the PDB50 collection of 1775 structures (with reapplication of the method to any domain of 250 and over) and the results were assessed, paying particular attention to assessing the generality of the errors observed above on the smaller data set and to any new problems that arose. 'Standard' graphs of number of domains and domain size were not compiled as these differed little from previous results and, as with other automatic methods, any differences are more a reflection of the constraints of the method rather than anything fundamental about proteins. The results of these calculations will be available in electronic form and, instead, some interesting examples are considered below. As a potentially rich source of domains, the 10 largest structures in the PDB50 data set were examined (Table V).

1jdbB (carbamoyl-phosphate synthetase), the largest protein, consists of a closed ring of mainly β/α-domains along with α-domains. Some of the former link across their β-sheets giving a high cumulated sheet-splitting error, such that the method failed to find a solution that did not break a β-sheet. Despite this, the best solution (87% agreement both across eight domains) is given in Table V.

1bglC (β-galactosidase) yielded two C-terminal domains on the first application (the larger of which did not resplit, despite having 2-fold pseudo-symmetry, because of a strong continuous β-sheet). The more tightly interacting remaining three N-terminal domains were obtained on reapplication of the method.

1kcw (ceruloplasmin) consists of three tightly packed large domains arranged around a pseudo-3-fold axis, each of which consist of two even more closely interacting cuprodoxin-like domains (Murphy *et al*., 1997). These levels were correctly parsed with the exception of the second domain, which was declared as one domain. Manual application of the method to this domain (smoothed with $r = 17$) obtained the expected split.

1rae (an aspartate transcarbamyolase) proved to be an unusual test for the method. This protein consists of two almost identical halves (related by a 2-fold axis); however, owing to inconsistent numbering in the PDB file, distant parts of the molecule (across both halves) appear to be linked. Despite this, the current method divided the structure into almost exactly matching pairs of domains regenerating the obscured symmetry.

1alo (aldehyde oxidoreductase) has a complex collection of closely interacting domains (that had not been parsed correctly before by any automatic method). The current method divided off two N-terminal domains, one of which was again divided but the other was less than 250 residues and so remained intact, despite having two clear domains. The remaining domain consists of three domains arranged in an obscure 3-fold which was revealed by the current method in two stages.

1dhx (adenovirus type-2 hexon) is probably one of the most 'messy' protein structures, with long 'unstructured' loops, some over 50 residues in length. Although undoubtedly vital to the virus coat, these loops obscure any regularity in the fold of the individual molecule. Usefully, the current method tends to remove such loops (as domains) as they have insufficient density for their evolved local label to convert (or be converted by) any neighbouring region. Two core β-domains were clearly uncovered.

1waj (DNA polymerase), like 1jdbB, is a ring of consecutive domains, all of which were cleanly divided, including the folded end of the coiled-coil extension as a small α-helix bundle.

1hkbB (hexokinase-I) has two clear domains and the obvious break-point was recognized by the current method. However, the first half also had a small α-domain cleaved off and the remainder underwent further sub-division into two similar halves. These sub-divisions, however, did not appear in the second domain, which remained undivided. Despite being

Table V. Domains in large proteins

PDB	len	predicted domains level 1	level 2	level 3
1jdbB	1057	[0:143,210:351] [144:208] [352:469] [554:665] [666:689,755:936,1037:1057] [690:754] [937:1051]	[0:116] [117:351]	
1bglC	1021	(3:540,552:613) (541:551,614:727) (728:1023)	(3:28,50:96,114:193,206:216) (29:49,217:331) (97:113,194:205,332:613)	
1kcw	1008	(1:338) (347:703) (704:1040)	(1:12,50:193) (13:49,194:338) [347:362,404:554] [363:403,555:703] (704:722,762:883) (723:761,903:940) (884:902,941:1040)	
1rae(A)	926	(1:141,289:310) (142:288) (100:151) (1:99)		
1rae(B)		(152:140,290:309) (310:99) (141:289) (100:153)		
1alo	907	(1:167) (183:333,362:461,486:496,521:531,823:842) (168:182,334:361,462:485,497:520,532:822,843:907)	(183:253,266:307,368:451) (254:265,308:367,452:842) (8022:579,751:761) (168:8021,580:750,762:907)	(168:182,623:766,796:807,856:870) (588:619,770:791,810:850,872:907)
1dhx	905	(44:84) (165:220,251:300) (152:164,221:250,301:332) (424:475) (85:151,333:405,498:524,549:662,698:725,937:967) (406:423,476:497,525:548,726:726,833:856) (663:697,727:832,857:936)		
1waj	903	(1:383) (384:394,414:485,554:594,668:681) (486:553) (396:411,597:663,683:767,877:903) (778:871)	(35:86,373:383) (1:34,87:104,352:372) (105:351)	
1hkbB	899	(28:302,378:465) (16:27,303:377) (466:914)	(68:220,445:465) (28:67,221:444)	
1dik	869	(2:244) (245:359) (373:518) (360:372,519:874)		
1qba	858	(28:187,547:566) (217:335) (188:216,336:546,567:885)		

The PDB identifier and the number of residues in the 10 largest proteins in the PDB50 data set are followed by their predicted domain definitions (specified as in Table VII). The method is reapplied to all domains of 250 and over and the results are given at each level of application. Definitions in square brackets, as distinct from parentheses, were obtained by manual intervention. See main text for details.

almost identical (weighted r.m.s.d. = 0.63, over 448 residues) this difference probably resulted from the different orientation of the N-terminal helix in the two main domains, and not from a context dependence, as the results remain different on the isolated domains. The situation is ideal for application of the multi-chain method and application of this results in an initial split of an α-domain, followed by a subdivision of the remainder into its two symmetric halves (weighted r.m.s.d. = 2.16, over 106 residues). Interestingly, the interface helices in this final split have been swapped (Heringa and Taylor, 1997), with each packing against the other half.

1dik (pyruvate-phosphate dikinase) and 1qba (chitobiase) are of interest as they both contain 8-fold β/α-barrels (TIM barrels). Given the propensity of these to split into two, it was encouraging to find that both had remained intact, even with the distinct bi-lobed form seen in 1dik. 1qba had a small β-domain left attached to the barrel but this is intimately linked by long loops and for all values of *r*, separation of this domain was associated with a split in the barrel resulting in no solution being acceptable within the allowed β-sheet disruption error.

The TIM barrel proteins mentioned by Jones *et al*. (1998) as being problematic (1brlB, 1btc and 1xyzA) were also examined and found to give the intact barrels as domains (in contrast to the other automatic methods assessed in Jones *et al*., 1998). These results, combined with the previous, suggest that over a large range of protein sizes, the somewhat *ad hoc* cutoff placed on β-sheet disruption (as the square root of the protein length) appears to be a reasonable constraint.

Sensitivity to chain breaks

With some of the larger proteins discussed above, and transmembrane proteins in general, the domain parsing was affected by breaks in the protein chain. While the simple approach is to avoid such proteins, this is not always a satisfactory solution. A better alternative would be to 'patch up' such defects by re-linking the broken chain ends; for example, a simple modification to the algorithm described in Methods could be used (replacing the centroid of the deleted segment by a reflection of the protein centroid through he mid-point of the chain break). However, because the domain definition method

Table VI. Effect of deletions on domain definition

delete	1	2	1	2
(native)	1–149	150–179	180–200	201–316
222–226	1–149	150–179	180–200	201–316
86–90	1–148	l49–180	181–200	201–316
107–111	1–148	l49–	———	–316
277–231	1–154	l55–167	168–203	204–316
12–16	1–154	l55–167	168–203	204–316

Groups of five residues were progressively deleted from the structure of thermolysin (4tln) and the changes in domain definition monitored. The protein has two domains (1 and 2) occurring mostly with two segments each.

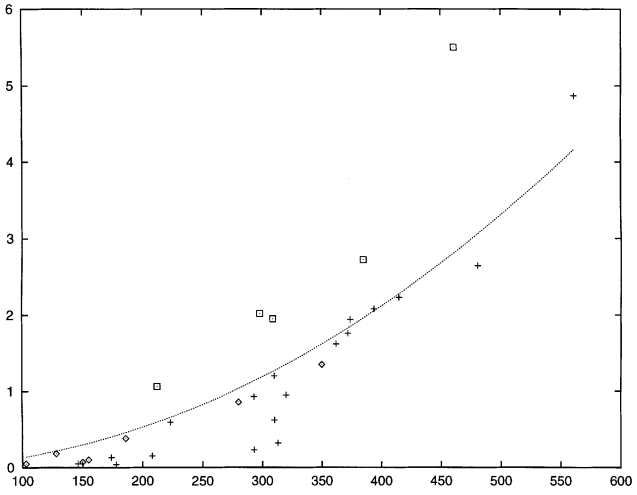


Fig. 2. Computation times. The elapsed computation time (seconds) is plotted against the number of residues for a variety of proteins: ◇, single-domain proteins; +, multiple-domain proteins; □, proteins which required more than one pass of the algorithm to resolve small fragments. The curve, represented by $(x/275)^2$, provides a rough estimation of computation time for medium-sized proteins.

does not explicitly make use of chain connectivity, it is not particularly sensitive to a few chain beaks (those mentioned above all involved six or more breaks). This can be illustrated by deleting loops from a protein model and reapplying the algorithm. Thermolysin (4tln) provides a good example, being an average sized double domain protein. It was chosen, however, because the domain definition is ambiguous and the current automatic definition differs from that in the ICRF server.

Deleting groups of five residues progressively from loop regions in the structure had little effect on the domain definitions up to three deletions (Table VI). With the fourth deletion, the loop previously part of domain 1 was claimed by domain 2 (equivalent to the ICRF definition). Further deletions led to a variant of the original allocation. This example illustrates that, while the method is sensitive to chain breaks, reasonable results continue to be generated even after 10% of the protein has been deleted.

Computation time

The computer program encoding the method was executed on a 400 MHz Pentium processor and the actual (elapsed) computation time recorded for a selection of single domain proteins and all the multiple domain proteins included in the UCL test set (under 600 residues). These values are plotted in Figure 2.

The computation times for a selection of larger proteins were also calculated (data not shown). These all involved at least double passes to resolve unassigned fragments and the times ranged from 18 s for 750 residues to almost 40 s for proteins over 1000 residues.

The program that implements the method will be made available on the ftp server at ftp://glycine.nimr.mrc.ac.uk/pub/.

Discussion

The current method, in its basic form, is one of the simplest that has been applied to the problem of domain definition, yet its predictions are acceptable for roughly 90% of proteins and deviate seriously from the accepted definitions only where the integrity of β-sheets is not preserved. Since the basic method has no inherent knowledge of protein structure, special treatment of these higher order structures cannot be expected, but its generality allows it to work on just α-carbon coordinates, without any pre-calculation of hydrogen bonding or solvent surface areas, and to have considerable tolerance to chain breaks (and erratically ordered PDB files).

To conform to expectations of unbroken β-sheets, a network of interactions within the β-structure was calculated. It was initially thought that these interactions could be accommodated as an extra term in the matrix of interatomic interactions (**P**, Equation 2). Although only partially tested, this solution was avoided as it appeared to have little influence on the results, with the same divisions being found, just with larger numbers being balanced. From a theoretical point of view, it also seemed preferable to have exactly the same model operating on all structural types of proteins For these reasons the β-bias was applied only to the starting conditions (the labelling) or as filters to the results (rejecting proteins with badly broken sheets), leaving the computational core untouched.

With the β-bias, the method was almost perfect across the UCL test data set, having only two errors, one of which was debatable (Table II, smooth structure with $r = 20$). One of these structures, however, was trypsin, which holds a special place in the history of the concept of a domain (McLachlan, 1979) that could not easily be ignored (with a clear conscience). Like the problem of black-body radiation in late nineteenth century physics, this small anomaly led to a more fundamental revision in the current approach to domain definition, as it was clear that the tightly packed trypsin domains are intuitively viewed with a finer level of granularity. As the level of domain granularity can be easily influenced in the current method, it was considered whether there might be some way in which an internally derived value might be found for this property that was optimal for each individual protein. A potential solution that has not been investigated in this work might be to introduce a degree of randomization into the evolution of the state labels. Repeated domain definitions with different starting configurations might then be used to determine the most stable domain assignment.

In the absence of any such fundamental solution, however, the more pragmatic solution of finding agreement between two different representations of the structure was developed in which the level of granularity was found that gave the best agreement between the predicted domains on the native and smoothed structure. While a good practical solution, this approach has theoretical limitations as there might be radically different solutions with trivially different levels of agreement. This problem was encountered above, where it was found that solutions involving fewer domains will be preferred as this

introduces less scope for differences. A more fundamental problem, however, is that the agreement between single domains is perfect and, if encountered, will be accepted as best. For these reasons, accepting the point of maximum agreement over a wide range of granularity was not a practical option and a search strategy was adopted beginning in the mid-range of granularity and searching outwards.

The combination of the basic method with the β-bias and variable granularity, developed on the small UCL sub-set of structures, gave good results when tested on the much larger ICRF50 data set. For the easier parsing problems, agreement was almost perfect (to within a residue or two) and for the majority of the more problematic proteins, the current method gave acceptable results that in many instances were an improvement over the recorded definition. Some protein families remained difficult, including (still) the trypsins, pepsins, lactate dehydrogenase and a few TIM barrel structures. Rather than continually refine the parameters of the method to try and encompass all of these difficult cases, a more detailed evaluation of the parameter space was made to see if there was any preferred direction in which to move. Looking at the joint influence of the granularity (r in Equation 2) and the β-sheet cutoff (h in Equation 4), however, revealed that the method was operating within reasonable bounds of these parameters and the only change introduced in the light of the ICRF50 results was to reintroduce the reapplication of the method to any domains of 250 or more residues.

The remaining difficult proteins were taken as test examples for the multiple structure extensions of the method to see if pairwise combinations of related structures would help resolve their domain definition. For the pepsins, the widely scattered single structure definitions converged on a three domain solution with a small β-linker domain being defined between the two commonly accepted domains. The trypsins showed some signs of movement towards a double domain but the lactate dehydrogenases remained with one domain, even with the recruitment of the related malate dehydrogenases (which did separate correctly into nucleotide binding and catalytic domains). It is probable that combinations of more structures, or simply more remote relatives, may help these difficult definitions but this aspect will be more completely investigated elsewhere when further problematic examples have been identified.

The final method was applied to a non-homologous structure collection and its behaviour evaluated on the largest of these proteins, which contain a wide variety of both clear and obscure domains. The method performed well, finding domains in proteins that had previously defined automatic definition. For a few proteins that contain related domains different results were obtained on each copy, but these were reconciled through application of the multi-structure version. These results suggest that the definition of structural domains and multiple protein structure comparison should proceed together in a concerted manner and future work will be made in this direction.

References

Adams,M.J., Ford,G.C., Koekoek,R., Lentz,P.J., Jr, McPherson,A., Jr, Rossmann,M.G., Smiley,I.E., Schevitz,R.W. and Wonacott,A.J. (1970) *Nature* **227**, 1098–1103.

Aszodi,A. and Taylor,W.R. (1993) *Comp. Appl. Biol. Sci*., **9**, 523–529.

Aszodi,A., Gradwell,M.J. and Taylor,W.R. (1995) *J. Mol. Biol*., **251**, 308–326.

Bangham,J.A. (1988) *Anal. Biochem*., **174**, 142–145.

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

Bruce,A. and Wallace,D. (1992) In Davies,P. (ed.), *The New Physics*. Cambridge University Press, Cambridge, pp. 236–267.

Feldman,R.J. (1976) *Atlas of Protein Structure on Microfiche*. Technical Report. Tracor Jitco, Rockville, MD.

Heringa,J. and Taylor,W.R. (1997) *Curr. Opin. Struct. Biol*., **7**, 416–421.

Holm,L. and Sander,C. (1994) *Proteins: Struct. Funct. Genet*., **19**, 256–268.

Islam,S.A., Luo,J. and Sternberg,M.J.E. (1995) *Protein Engng*, **8**, 513–525.

Janin,J. and Chothia,C. (1985) *Methods Enzymol*., **115**, 420–440.

Janin,J. and Wodak,S.J. (1983) *Prog. Biophys. Mol. Biol*., **42**, 21–78.

Jones,S., Stewart,M., Michie,A., Swindells,M.B., Orengo,C. and Thornton,J.M. (1998) *Protein Sci*., **7**, 233–242.

McLachlan,A.D. (1979) *J. Mol. Biol*., **128**, 49–79.

Murphy,M.E.P., Lindley,P.F. and Adman,E.T. (1997) *Protein Sci*., **6**, 761–770.

Phillips,D.C. (1966) *Sci. Am*., **215**, 78–90.

Rashin,A. (1985) *Methods Enzymol*., **115**, 420–440.

Rose,G.D. (1979) *J. Mol. Biol*., **234**, 447–470.

Rossmann,M.G., Moras,D. and Olsen,K.W. (1974) *Nature*, **250**, 194–199.

Sali,A., Veerapandian,B., Cooper,J.B., Moss,D.S., Hofmann,T. and Blundell,T.L. (1992) *Proteins: Struct. Funct. Genet*., **12**, 158–170.

Siddiqui,A.S. and Barton,G.J. (1995) *Protein Sci*., **4**, 872–884.

Sowdhamini,R., Rufino,S.D. and Blundell,T.L. (1996) *Fold. Des*., **1**, 209–220.

Swindells,M.B. (1995a) *Protein Sci*., **4**, 103–112.

Swindells,M.B. (1995b) *Protein Sci*., **4**, 93–102.

Taylor,W.R. (1987) *Comp. Appl. Biol. Sci*., **3**, 81–87.

Taylor,W.R. (1998) *J. Mol. Biol*., 280, 375–406.

Thouless,D. (1992) In Davies,P. (ed.), *The New Physics*. Cambridge University Press, Cambridge, pp. 209–235.

Zimm,B.H. and Bragg,J.R. (1959) *J. Chem. Phys*., **31**, 526–535.

Received August 18, 1998; revised November 16, 1998; accepted November 26, 1998

Appendix

Comparison with the ICRF50 definitions

The full method described in the main text (searching outwards from $r = 14$ for agreement between the native and smooth structures with the β-sheet bias and filter) was applied to the ICRF50 collection of structures and the predicted domain definitions compared to those on the server (see Methods for details). For analysis, the results can be divided into single- and multiple-domain proteins, and those for which no agreement was obtained.

No solution found. Of the 517 proteins considered, only six failed to reach agreement. These were generally tightly packed domain pairs, often related by internal symmetry and cross-linked through a β-sheet. Failure to find a solution stemmed from the opposing drives to maintain the integrity of the β-sheet and split what are mostly clearly bi-lobed structures.

Calculating the domains with a single pass using the smoothed structure and a neighbour cutoff radius of $r = 20$ provided good solutions for all the proteins (with the exception of 2tmaA, which is a single long helix) (Table VII).

Single-domain proteins. The predictions of the current method agreed substantially with the ICRF definitions, with only 10% (roughly) of the proteins defined as one domain being predicted to have more than one. However, examination of these assignments in detail revealed that many of the predictions were acceptable or differ in minor or trivial aspects (Table VIII).

Of the 24 'single' domains predicted as multiple, only two can be said to be clearly wrong. These were both TIM barrel (8-fold β/α) proteins, one of which (3chb) was split in two as a result of a weak link in the hydrogen bonding around the β-sheet, while the other (1nar) is a simple barrel with good hydrogen bonding. Some further structures had ambiguous divisions: these were 1ayaA, which has two β-sheets, each of which can form the basis of a domain but the resulting fragments were rather small, and 2scpA, which was split into its component EF hands (calcium binding motifs). 2asr is a four-helix coiled coil which should probably be best left as one domain, while 1rveB has been discussed above.

Multi-domain proteins predicted as one. Thirty-nine proteins defined as multi-domain on the ICRF server were predicted as single domains. These are analysed in detail in Table IX. Six of these domains were too small to be recognized by the algorithm (<40 residues). Typical among these were zinc-finger domains. Of the remainder, 15 should have been divided into domains; however, this figure is inflated by distant homologues and includes three lactate dehydrogenases (1ldnB, 1lldA and 6ldh) and three trypsins. Both these families, along with the acid protease (1pplE) and the periplasmic-binding protein (1pbp), have other relatives that were correctly predicted.

Multi-domain predictions. Unlike the previous categories, multi-domain predictions can be compared directly with multi-domain definitions and a percentage accuracy calculated. Over the remaining 124 proteins this value was 90.6%. Although acceptable, as an average this figure hides many examples where the domain definitions were substantially different. Thirty-nine non-trivial differences were identified, ranging from the different allocation of loops to different numbers of domains. These are analysed in Table X where it can be seen that most of the predicted domain definitions are acceptable

Table VII. Proteins for which the smooth and native definitions did not agree

PDB	ICRF definition	predicted domains	comments
2tmaA	ONE		single very long helix
1mpp	(1:173)(174:325)	(13:147)(172:332)	aspartyl protease
2apr	(3:177)(178:323)	(1:155)(156:325)	aspartyl protease
4gpd1	(1:148)(149:333)	(1:117)(118:333)	glycerol dehydrogenase
1tbpA	(61:71,159:240)(68:158)	(61:68,160:240)(69:159)	TATA binding protein
2polA	(1:121)(121:249)(250:366)	(1:120)(121:246)(247:366)	POL III (beta subunit)

Each PDB code is given followed by its ICRF domain definition and the definition predicted for the smooth structure with $r = 20$. Domains are specified in the form 1abc > (1:5,31:50)(6:30,51:70), meaning that the first domain (in parentheses) consists of segments 1–5 and 31–50 and the second domain (again in parentheses) includes segments 6–30 and 51–70.

Table VIII. Single-domain proteins predicted as two or more

PDB	predicted domains	status	comments
1brd	(8:60)(61:225)	OUT	transmembrane with no loops.
1prcC	(1:7,23:141,310:332)(8:22,142:309)	OUT	transmembrane with no loops.
1prcM	(10:187)(188:211,267:305)(1:9,212:266)	OUT	transmembrane with no loops.
4rcrM	(6:188)(189:213,269:301)(214:268)	OUT	transmembrane with no loops.
1prcL	(1:39,98:118)(40:97,119:159)(160:185,231:273)(186:230)	OUT	transmembrane with no loops.
1hfh	(1:63)(64:120)	OK	two clear domains
1hgiB	(1:23,36:60,92:119)(24:35,120:175)	OK	C.coil split off
1hvc	(10:94)(1:9,95:99)	OK	2 clear domains (D-protease, linked dimer)
1cpt	(1:98,315:353)(99:314,354:428)	OK	prob. 2 (B/A domain and large A-domain can split)
1eriA	(17:114,142:170,201:277)(115:141,171:200)	OK	probable two (C-term. double loop extension split off).
1rveB	(2:171)(172:245)	OK?	possible two. (C-term. alpha extension split off).
1ximA	(3:324)(325:394)	OK	TIM-barrel + long Cter coil (latter split off)
2bpa1	(1:164,212:292,384:426)(165:211,293:383)	OK	B-barrel + extensive loops (latter split off)
2pf2	(1:61)(62:145)	OK	small B-domain + end loops (latter split off)
2plv1	(6:73)(74:302)	OK	large B-domain + unstructured N-term. loops (split off)
2scpA	(13:79)(1:11,110:174)	OK?	possible split into two EF-hand sub-domains
3gapB	(1:128)(129:205)	OK	two clear domains
3blm	(92:138)(31:91,139:290)	OK	small insert cut-out
7apiA	(20:191,293:358)(192:292)	OK	probable two (same as 1hleA)
1hleA	(23:191,293:358)(192:292)	OK	probable two (same as 7apiA)
3cbh	(83:101,134:262,284:298)(102:133,263:283,299:447)	NO	TIM-barrel (weak N-C ter link in sheet not made)
1nar	(1:189)(192:212,224:255,267:289)	NO	TIM-barrel split
1ayaA	(3:56,97:103)(57:96)	NO?	split OK but too small (101)
2asr	(38:111)(112:179)	NO?	long 4-fold C.coil split into two C.coil hairpins

Each PDB code is given followed by its predicted domain definitions (see Table VII for details). All these proteins were defined as single domains at the ICRF domain server. The comment OUT indicates that the protein should have been omitted from consideration and OK indicates that the predicted domain is either clearly correct or within the limits of ambiguity. A NO indicates that the predicted domain definition is unacceptable (a ? appended to these indicates some uncertainty in the assignment). B and A are used as abbreviations for β and α and TIM barrel indicates an 8-fold alternating β/α barrel.

Table IX. Multiple domain proteins predicted as one domain

PDB	ICRF domains	status	comments
1bbo	(1:25)(29:57)	NOT	2 Zn fingers (too small)
1esl	(1:120)(121:157)	NOT	small Cterm B dom. (To small to declare)
1bn21	(1:58)(59:86)	NOT	2 small B domains.
2drpA	(103:138)(139:165)	NOT	2 Zn.fingers
4mt2	(1:31)(32:61)	NOT	2 small domains (less than limit)
1zaaC	(3:31)(32:60)(61:87)	NOT	3 Zn.fing doms (below limit)
1pfc	(334:409)(415:441)	OK	single Ig.dom
1ede	(1:155,230:310)(156:229)	OK?	long A hairpin split off B/A core (but packed tight). Same as 2had
3cox	(5:44,226:316,462:506)(45:225,317:461)	OK?	possibly two but very tight interactions
1lybB	(106:189)(190:345)	OK	split good sheet
1abmA	(1:84)(85:198)	OK?	C.coil split off
1cauA	(44:177)(178:224)	OK	Cter loop cut (but back-links to sheet). Same as 1cauB
1cauB	(241:379)(380:424)	OK	Cter loop cut (but back-links to sheet). Same as 1cauA
1cpcA	(1:34)(35:174)	OK	Nter A-hairpin cut OK but better left. Same as 1cpcB.
1cpcB	(1:35)(36:174)	OK	Nter A-hairpin cut OK but better left. Same as 1cpcA.
1grcA	(1:100)(104:209)	OK	split sheet
1emd	(1:148)(149:309)	OK?	tight interaction and sheet split (at weak point)
1mat	(11:118)(119:241)	OK	split good sheet
1ahc	(1:181)(182:246)	OK?	small Cter dom cut
1iag	(2:149)(150:202)	OK?	small Cter dom cut (not compact)
1dsbA	(1:62,139:188)(63:138)	OK?	A insert dom cut
4icd	(3:124,318:416)(125:157,203:317)(158:202)	OK	split good sheet. Same as 1ipd.
1ipd	(1:90,252:344)(91:120,159:251)(121:158)	OK	split good sheet. Same as 4icd.
1gal	(3:56,228:323,521:583)(56:228)(324:520)	OK?	tight interactions + split sheet.
2sga	(16:126)(127:242)	NO	trypsin (2 B-barr)
2alp	(15A:122)(123:242)	NO	trypsin (2 B-barr)
1arb	(1:139,229:263)(140:228)	NO	trypsin-like but with strong domain X-links
1ldnB	(15:164)(165:330)	NO	di.nuc bind dom packs tightly to Cter dom (as 1lldA and 6ldh)
1lldA	(7:151)(152:319)	NO	di.nuc bind dom packs tightly to Cter dom (as 1ldnB and 6ldh)
6ldh	(20:164)(165:329)	NO	di.nuc bind dom packs tightly to Cter dom (as 1lldA and 1ldnB)
1pplE	(5:173)(174:322)	NO	D.protease
1pbp	(1:78,210:321)(79:209)	NO	periplasmic-binding fold but with strong domain X-links
1chrA	(1:122)(130:327)(339:367)	NO	TIM-barrel + big N-term. A dom (packed) + bit on C-term.
1mioA	(2:51,322:521)(52:203)(204:321)	NO	tight interactions + long packed loops .
2dkb	(3:49)(50:325)(326:433)	NO	packed but clear doms.
1fcdA	(1:107,256:328)(108:255)(328:401)	NO	2 clear but odd 3rd
1tnrR	(15:52)(53:97)(98:137)(138:153)	NO	tight linked (egf-like) domain string
1pxtA	(28:153,276:298)(154:275)(299:417)	NO?	tight BA domains (ABABA layers) 2 doms with symm
2snv	(114:177)(180:264)	NO?	close B on B packing

Each PDB code is given followed by its domain definitions from the ICRF server. These are specified as in Table VIII. The status NOT indicates that the domains were too small to be declared by the algorithm (<40) and OK indicates that the predicted single domain is either clearly correct or within the limits of ambiguity. A NO indicates that the predicted single domain definition is unacceptable (a ? appended to these indicates some uncertainty in the assignment).

and some even preferable to those on the ICRF domain server. When mean agreements were calculated separately over both sub-sets, a value of 97.6% was obtained over the 85 uncontentious definitions (the lowest of which was 82.7%), indicating almost exact agreement, whereas over the contentious sub-set, the mean dropped to 75.4%.

The differences collected in Table X have been split into categories of severity. Four assignments (under ICRF error) are probably typographical or possibly associated with different residue numbering. Seven predicted definitions are clearly better than those on the server and a further five involve more minor loop reassignments which can also be considered to be an improvement. Of the remaining 23 proteins, nine are ambiguous. Among these were actin (1atnA, referred to in Table III) in which the main (pseudo-symmetric) split into two domain was found but no further sub-division. Pepsin again emerged as ambiguous, being split into three domains rather than the conventional two. However, the third domain constitutes an interface sheet that can be considered as a separate domain (Sali *et al.*, 1992). 5-Enol-pyruvyl-3-phosphate synthase (1eps) gives an interesting example of multi-layered domains: while clearly dividing into two domains, one half can be split into two symmetric parts and the other into three. The prediction settles at a level keeping the amino-terminal domain intact but reapplication of the algorithm to this fragment allows the full split into thirds. Similarly, the nitrogenase 1mioB is split into two and then one of the domains resplit into symmetric (flavodoxin-like) halves. 1gsgP encounters the same parsing level difference with an equivalent result being obtained after two applications.

A further 10 domain definitions differ in this way, and although there is no clear line dividing right from wrong, the ICRF definitions were considered preferable for these proteins. Some of these involved splits that might have been better left, such as that in 2glsA (glutamine synthetase) in which a weak point is found to break the β-sheet, or 3gly (glucoamylase) in which a large ring of α-helices is split into pseudo-symmetric halves. Others are failures to split small, tightly packed, domains away from larger neighbours [for example the C-terminal domain of 1gof (galactose oxidase), which inserts a 'finger' deep into the centre of the larger 7-fold propeller domain]. Only three predicted definitions can be clearly said to be wrong. These included 2at2A, which is a homologue of the aspartate transcarbamolyase included in the UCL sub-set (8atcA) and in which the same error is made (see above). A pair of helices is similarly split off in 1rpa but there seems to be no obvious reason why these small domains should have been split off. The final error in 1lla (haemocyanin) involved a bad split through the middle of the central domain; however, this protein contains several chain breaks in this region and, like the transmembrane proteins, should not have been presented to the algorithm.

Table X. Differences in multi-domain definitions

PDB	ICRF domains	status & comments
	——— ICRF error ———	
1mcoH	(1:320)(326:426)	OK 1st 3 doms not split by ICRF!
	(1:118)(119:219)(220:323)(324:428)	
1glbG	(5:253)(254:4)	OK? typo at ICRF?
	(4:246,439:456)(304:391,404:436,465:499)[(304:8088,477:499)(8089:476)]	
2aaa	(1:100,169:407)(101:168)(408:496)	OK poss. typo in ICRF file?
	(1:366)[(1:230,342:354)(231:341,355:366)](367:476)	
1tplA	(19:48,333:456)(49:56,311:332)(57:310)	OK non-dom.2 at ICRF!
	(51:319)(1:50,320:456)	
	——— DOMS better ———	
1rne	(1:173)(174:325)	OK N-term. strand belongs in dom.2
	(16:144)(-1:15,145:326)	
2phlA	(11:148)(149:279)	OK 2 bits crossed to dom.2 (one is sheet strand)
	(22:36,53:220)(11:21,37:52,221:381)	
1gph1	(1:230)(231:456)	OK Cterm assigned to dom.1
	(1:254,427:465)[(1:234,445:465)(235:444)](255:426)	
1sesA	(1:98)(99:421)	OK Ccoil loop split off
	(37:90)(1:36,91:102)(103:421)	
1tde	(1:116)(117:244)(245:316)	OK bad 3rd split by ICRF
	(1:116,245:316)(117:244)	
2mnr	(3:126)(134:319)(331:359)	OK TIM + other (split by ICRF)
	(3:17,31:120,347:359)(18:30,121:346)	
7catA	(3:75)(76:320)(321:436)(437:500)	OK 3 dom better small A dom more complete
	(11:66)(3:10,67:152,201:424)(153:200,425:500)[(153:175,8051:434)(176:8050,435:500)]	
	——— loop difference ———	
1phh	(1:74,97:181,269:391)(75:96,182:268)	OK Cterm As assigned to 1st domain
	(1:67,102:178,271:338,389:394)[(1:157)(158:394)](68:101,179:270,339:388)	
1gpb	(19:489)(490:841)	OK small 'tower' dom cut out
	(19:57,106:120)(58:105,121:166,179:485,813:841)[(58:202,218:338,483:841)(203:217,339:482)](167:178,486:812)	
6tmnE	(1:135)(137:316)	OK alternate split keeping loop with domain where it packs
	(1:149,179:200)(150:178,201:316)[(150:238)(239:316)]	
3grs	(18:157,294:364)(158:293)(365:478)	OK C.coil goes with dom.2
	(18:60,109:159,292:364)(61:108,160:222,235:291)[(61:8053)(8054:291)](223:234,365:478)	
1tytB	(2:160,289:358)(161:288)(359:471)	OK C.coil loop goes with dom.3
	(2:57,101:164,287:360)(58:100,165:286)[(58:8046)(8047:286)](361:487)	
	——— ambiguous ———	
2hpdA	(1:70,329:361)(72:325,390:457)	NO? but no clear splits
	(1:108,164:265,328:369,385:402)[(1:71,8225:402)(72:79,8115:211)(80:8114,212:8224)](109:163,266:327,370:384,403:457)	
4pep	(1:173)(174:325)	OK? linker sheet forms 3rd dom
	(16:146)(1:15,147:182,307:326)(192:306)	
1eps	(1:19,239:427)(20:238)	OK dom.2 split in two (dom.1 resplit in 3 symm. parts)
	(20:240)[(20:79,232:240)(80:158)(159:230)](9:19,241:301,410:427)(1:8,302:409)	
1dpi	(326:517)(520:928)	OK finer split in middle dom.
	(326:516)(531:693)(517:530,694:708,851:928)(709:850)[(709:799)(800:850)]	
1hgeA	(1:94,260:328)(95:259)	OK? ambiguous in both
	(13:40,315:328)(41:314)	
1hsbA	(1:90)(91:182)(183:270)	OK MHC anti-bind + Ig ab.dom not split (good sheet)
	(1:181)(182:270)	
1mioB	(22:160)(171:284)(313:458)	OK but dom.1/2 on resplit
	(22:290)[(22:153,204:223)(154:203,224:290)](2:21,291:458)	
1atnA	(1:32,70:144)(33:69)(145:180,270:337)(181:269)	OK 2 main doms found
	(1:137,339:372)(138:338)	
1gsgP	(8:100,211:260)(101:210)(260:339)(340:348,465:547)(349:464)	OK dom.1 and 3 resplit to give same
	(8:260)[(8:98,220:233,252:260)(101:216,234:250)](261:331,472:496)(332:471,497:547)[(347:367,379:463)(332:346,368:378,464:547)]	
8acn	(2:200)(201:317)(320:513)(538:754)	OKish dom.1 is better not split?
	(2:100,120:319,515:540)(101:119,320:514)(541:754)	
	——— ICRF better ———	
2glsA	(1:102)(103:468)	OK? but 3 doms splits sheet
	(1:103)(104:104,130:266)(105:129,267:468)	
3gly	(1:440)(441:471)	OK? AA barr. split in 2 by DOMS
	(16:224,436:471)(1:15,225:435)	
3mddA	(11:128,254:395)(129:253)	OK? Nterm 4A bund. split (then resplit as Bbarr + A)
	(11:254)[(11:98)(99:254)](255:395)	
1aozA	(1:123)(130:317)(337:524)	NO 3 dom.s (but 2+3 got on resplit)
	(1:66,82:327)[(1:128)(129:327)](67:81,328:552)	
3ladB	(1:150,280:348)(151:279)(349:462)	OKish small 4th dom 'steals' part of C-term. dom. (as 1lvl)
	(1:52,97:153,279:349)(53:96,154:215,229:278)[(53:8048)(8049:278)](363:472)	
1lvl	(1:142,268:335)(143:267)(336:449)	OKish small 4th dom 'steals' part of C-term. dom. (as 3ladB)
	(1:45,96:145,267:336)(47:95)(147:265)(345:458)	
1trkA	(3:322)(323:538)(539:680)	OKish poor split in middle
	(3:129,148:311)(130:147,312:336,424:434)(337:342,367:423,435:464)(343:365,467:680)	
1gof	(1:152)(153:532)(542:639)	OKish but small Cterm dom not split from TIM
	(1:152)(153:639)	
1hkg	(2:24,253:370)(24:50,191:252,371:432)(51:190,433:458)	NO? 3 better (but dom.2 resplit)
	(15:286,361:458)[(56:188,432:458)(15:49,203:429)](2:14,287:360)[(2:316)(317:360)]	
1tahB	(2:117,166:213,272:319)(118:165)(214:271)	OKish 2 inserts only one split by DOMS
	(2:210,266:280,303:319)(211:265,281:302)	
	——— DOMS error ———	
2at2A	(1:130,273:295)(137:268)	NO 2 dom.s is better
	(35:121)(1:34,122:132,275:295)(133:274)	
1rpa	(1:126,227:342)(127:226)	NO a few helices form bad dom.2
	(16:37,150:208)(13:15,38:67,81:109)(1:12,68:80,110:149,209:342)	
1lla	(2:120)(155:380)(381:628)	NOish some poor splits (chain has many breaks)
	(2:147,413:433,514:533)[(2:111,517:533)(112:516)](150:191,226:269,324:362,411:412,434:435,513:513,534:536)(192:210,223:225, 270:323,363:378,410:410,436:436,508:512,537:541,567:595)[(192:-)(-:595)](211:222,379:409,437:507,542:566,596:628)	

As Table VIII but each protein has both the ICRF (top line) and the predicted definitions (second line). The latter also includes the result of applying the method to the separated domains (in square brackets). Residue numbers over 8000 are in artificial loops.