Protein Science (1995), *4*:872-884. Cambridge University Press. Printed in the USA.

Continuous and discontinuous domains: An algorithm for the automatic generation of reliable protein domain definitions

ASIM S. SIDDIQUI AND GEOFFREY J. BARTON

Laboratory of Molecular Biophysics, University of Oxford, The Rex Richards Building, South Parks Road, Oxford OX1 3QU, United Kingdom

(RECEIVED December 14, 1994; ACCEPTED February 17, 1995)

Abstract

An algorithm is presented for the fast and accurate definition of protein structural domains from coordinate data without prior knowledge of the number or type of domains. The algorithm explicitly locates domains that comprise one or two continuous segments of protein chain. Domains that include more than two segments are also located.

The algorithm was applied to a nonredundant database of 230 protein structures and the results compared to domain definitions obtained from the literature, or by inspection of the coordinates on molecular graphics. For 70% of the proteins, the derived domains agree with the reference definitions, 18% show minor differences and only 12% (28 proteins) show very different definitions. Three screens were applied to identify the derived domains least likely to agree with the subjective definition set. These screens revealed a set of 173 proteins, 97% of which agree well with the subjective definitions.

The algorithm represents a practical domain identification tool that can be run routinely on the entire structural database. Adjustment of parameters also allows smaller compact units to be identified in proteins.

Keywords: automatic domain definitions; contacts; domains database; protein structural domains

The concept of the domain has long been convenient to simplify and classify protein structure. Although there is no strict, universally accepted definition of a domain, domains are normally considered to be compact, local, semi-independent units (Richardson, 1981). In a multidomain protein, the domains may make up functionally and structurally distinct modules (Campbell & Baron, 1991; Baron & Campbell, 1991). Modules are usually formed from a single continuous segment of protein chain (Fig. 1A), and it is conceptually easy to see how such domains with similar three-dimensional structures may have arisen in different proteins by exon shuffling (Patthy, 1994). However, examination of multidomain proteins also reveals compact regions that are built of two or more nonsequential segments as illustrated in Figure 1B and C and Kinemages 1 and 2 (Russell, 1994). Although domains can be identified subjectively by eye, their importance to protein architecture and their possible role as independent nucleation sites in protein folding (Wetlaufer, 1973) prompted several groups during the late 1970s and early 1980s to investigate more systematic techniques for domain identification.

Reprint requests to: Geoffrey J. Barton, Laboratory of Molecular Biophysics, University of Oxford, The Rex Richards Building, South Parks Road, Oxford OX1 3QU, UK; e-mail: geoff@biop.ox.ac.uk.

Rossmann and Liljas (1974) applied Phillips-Ooi Cα–Cα distance maps (Phillips, 1970; Nishikawa et al., 1972; Nishikawa & Ooi, 1972) to locate domains. They suggested that a domain has many short residue–residue distances within itself, but few short distances with the rest of the protein. Although a powerful abstraction, distance plots require human interpretation. In an attempt to automate the identification of domains, Crippen (1978) applied hierarchical cluster analysis to protein fragment/fragment contacts. This procedure generated a hierarchical tree of protein fragments from small, locally compact regions through to the complete protein. Rather than build up from fragments, Rose (1979) examined the complete protein to find the optimum point to cut the polypeptide chain based on the geometry of the protein. The procedure generated a hierarchy of fragments but was only able to deal with single segment (continuous) domains. Instead of considering cutting planes or simple distances, Wodak and Janin (1981) calculated the interface area between two segments of the protein. They chose the minimum in the interface area as the domain boundary. The approach was extended to deal with domains made of two segments, though this was computationally expensive and not fully automated. Rashin (1981), Go (1983), and Zehfus and Rose (1986) applied globularity or compactness as domain definitions, but their methods

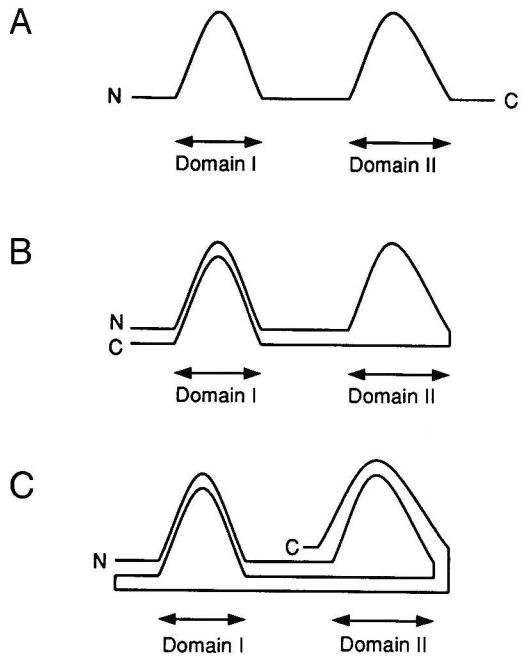

Fig. 1. Schematic diagram showing three possible paths that the polypeptide chain may follow in a two-domain protein. **A:** The simple case in which the chain first passes through one domain and then the other. **B:** The chain runs from the first domain into the second and then back into the first to complete it. **C:** Same as B except that, after the chain completes the first domain, it passes back into the second to complete it.

could deal only with single segment domains. More recently, Zehfus (1994) used compactness as a measure of "domainness" and searched for compact units in the structure composed of two noncontiguous sections of the chain. The technique resulted in a series of overlapping domain units, but did not provide a unique definition of the domains in the protein. Furthermore, the method could not be run in a reasonable time on proteins that contained more than 300 residues. Holm and Sander (1994) describe a method that searches for potential folding units using an eigenvalue analysis of contact maps. Although their elegant and fast method deals with multiple segment domains, many of their published domain definitions disagree with those found in the literature.

With the current rapid growth in the number of known protein three-dimensional structures, there is a pressing need to identify systematically the domains. Knowledge of domain locations is important in any reference database of protein structure, such knowledge is also needed for construction of representative sets of protein structures for derivation of parameters in prediction. Prediction of protein structure by threading techniques (Jones et al., 1992; Bowie & Eisenberg, 1993; Bryant & Lawrence, 1993; for review see Wodak & Rooman, 1993) is best approached at the domain level because this reduces the computational overhead. Furthermore, if effective methods are to be developed to identify domain boundaries in proteins of unknown three-dimensional structure, then a reliable library of domains is required to derive the necessary parameters.

A problem faced by all methods of domain definition is how to assess the quality of the domains that are identified. The majority of the early techniques reviewed above apply a simple physical or geometric model to divide a protein into domains. Although domains defined in this way may provide new insights about the protein structure, they do not always agree with the domain definitions in the literature. Accordingly, the approach adopted in this paper is to start from a subset of known protein structures for which the domain definitions have been well established, then derive a method that can reproduce the definitions automatically. The success of the method is evaluated by application to a larger test set of proteins. A domain reference set has been constructed from domain definitions described in the literature. Where definitions for a protein have not been described, assignments have been made by inspection. The new method starts from a simple geometric model similar to that used by Wodak and Janin (1981) (a domain has more residue-residue contacts within than without). However, alone this is insufficient to reproduce the normally accepted domain boundaries. The method has been refined to take into account secondary structure content and other factors in order to improve the agreement with the training set. Finally, three simple rules that are applied to any domain definition obtained by the method provide a ranking scheme to identify the definitions that are most likely to be correct.

The method explicitly allows for two-segment domains and implicitly allows the formation of three- or more segment domains. It runs in a reasonable time on proteins of any size and can optionally provide a hierarchical classification of compact regions within the protein.

A unique definition of the domains is presented for a set of 230 protein chains. Automatic screening of this set picked out 173 proteins, of which 97% agreed with the reference definitions.

Results

Comparison of domain definitions

Table 1 shows definitions of the domains found by the program DOMAK with default parameters. Table 1 also illustrates the corresponding reference definitions obtained from the literature and visual inspection (see Materials and methods). In the following discussion the set of definitions obtained by the algorithm is referred to as the derived set.

For 161 of the proteins (Set A), the derived domains agree with those in the reference set (see Materials and methods section for definition of reference set). This gives a confidence level of 70% for the method. Only 28 proteins (12%) (Set C) had all domains defined differently to the reference set.

Domain definitions for 41 proteins (18%) (Set B) did not agree closely with the reference domains, but either had one or more identically defined domains, or by inspection were split into what one would subjectively term domains. The 41 proteins in Set B highlight some of the difficulties with subjective definitions of domains. For example, glycogen phosphorylase is split into two domains (Kinemage 2). However, 18 residues at the C-terminus come back across the N-terminal domain. As the tail packs loosely against the first domain, the reference definitions do not assign it as part of either domain. However, DOMAK assigns it to the C-terminal domain. A further example is actin, which the authors of the structure classed as having two domains (Kinemage 3; Kabsch et al., 1990). The first domain consists of residues 1-144 and 338-375 (domain I in Fig. 2) and the second domain of residues 145-337 (domain II in Fig. 2). However, it

Table 1. *Domains found by DOMAK*[a]

A	B	C	D	E	F	G
‡1aai†	A	1	1	1-267	3	1-117
			2			118-210
			3			211-267
1aai†	B	3	1	188-257	2	1-135
			2	139-187, 258-262		136-262
			3	1-138		
‡1aak	–	1	1	1-150	1	ALL
‡1aap	A	1	1	1-56	1	ALL
‡1aaq	B	1	1	10-99	1	ALL
‡1aar	A	1	1	1-76	1	ALL
‡1aba	–	1	1	1-87	1	ALL
1ace	–	3	1	4-315	1	ALL
			2	346-399, 522-534		
			3	316-345, 400-484,		
			–	490-521		
1ake	A	2	1	1-111, 174-214	1	ALL
			2	112-173		
1alc	–	2	1	38-104	2	38-104
			2	1-37, 105-122		1-37, 105-122
‡1ald	–	2	1	191-307, 342-363	1	ALL
			2	1-190, 308-341		
1ama†	–	3	1	74-296	2	48-325
			2	48-73, 297-324		15-47, 326-410
			3	13-47, 325-410		
‡1apk	–	1	1	143-260	1	ALL
‡1aps	–	1	1	1-98	1	1-98
‡1atn†	A	3	1	1-137, 358-372	2	1-144, 338-375
			2	138-185, 272-357		145-337
			3	186-271		
‡1atn	D	2	1	89-225	1	ALL
			2	1-88, 226-260		
1avr†	–	3	1	161-225	4	14-86
			2	3-14, 86-160,		87-160
				226-304		
			3	15-85, 305-318		161-346
			4			247-318
‡1bbh	A	1	1	1-131	1	ALL
‡1bbk	A	6	1	120-204	1	ALL
			2	205-253		
			3	29-68		
			4	69-119		
			5	254-320		
			6	321-373		
‡1bbk	B	1	1	16-131	1	ALL
‡1bbp	A	1	1	2-178	1	ALL
‡1bbt	1	1	1	24-192	1	31-189
‡1bbt	2	1	1	9-218	1	ALL
‡1bbt	3	1	1	52-220	1	42-214
‡1bia†	–	3	1	1-64	3	1-60
			2	65-270		61-271
			3	271-317		272-317
1bic	–	2	1	149-173, 217-241	1	ALL
			2	3-148, 174-216,		
				242-261		
‡1bmv	1	1	1	1001-1185	1	ALL
‡1bmv	2	2	1	3012-3181	2	3025-3181
			2	3182-2192		3182-2189
‡1bov	A	1	1	1-69	1	ALL
‡1bpk	–	1	1	261-379	1	ALL
‡1brd	–	1	1	8-225	1	ALL
‡1caa	–	1	1	1-53	1	ALL
1cbx	–	2	1	1-127, 174-307	1	ALL
			2	128-173		
‡1cc5	–	1	1	5-87	1	ALL
‡1cd8	–	1	1	1-114	1	ALL
‡1cho	E	2	1	1-27, 122-235	2	1-16, 124-233
			2	28-121, 236-245		28-123, 234-245
‡1cho	I	1	1	4-56	1	ALL
‡1cmb	A	1	1	1-104	1	ALL
1cob	A	2	1	43-83, 114-146	1	ALL
			2	1-42, 84-113,		
				147-151		
‡1col	A	1	1	5-201	1	ALL
1cox†	–	4	1	5-43, 216-286	2	45-225, 317-461
			2	156-215, 324-381,		5-44, 226-316,
				412-441		462-506
			3	95-155, 287-323,		
				382-411, 442-506		
			4	44-94		
‡1cpc	A	1	1	1-174	1	ALL
1csc†	–	2	1	1-282, 378-423	2	1-274, 381-437
			2	283-377		275-380
‡1cse	I	1	1	8-70	1	ALL
‡1ctf	–	1	1	53-120	1	ALL
‡1cwg	A	4	1	2-42	4	2-41
			2	43-85		42-88
			3	86-130		89-128
			4	131-171		129-171
1drf	–	2	1	1-5, 38-116	1	ALL
			2	6-37, 117-186		
‡1eca	–	1	1	1-136	1	ALL
‡1end	–	1	1	2-138	1	ALL
‡1epi	–	1	1	1-53	1	ALL
‡1etu	–	1	1	5-200	1	ALL
‡1ezm†	–	2	1	1-142, 171-199	2	1-135
			2	143-170, 200-298		136-298
‡1fha	–	1	1	6-183	1	ALL
‡1fia	A	1	1	48-98	1	ALL
‡1fkb	–	1	1	1-107	1	ALL
‡1flx	–	1	1	1-79	1	ALL
‡1fnr	–	2	1	19-24, 153-314	2	19-152
			2	25-152		153-314
‡1fxd	–	1	1	1-58	1	ALL
‡1fxi	A	1	1	1-96	1	ALL
‡1gal	–	2	1	67-98, 176-210,	3	3-108, 225-327
				323-517		514-583
			2	3-66, 99-137,		109-225
				211-322, 518-583		
			3			328-513
‡1gd1	O	2	1	0-147, 314-333	2	0-148, 318-333
			2	148-313		149-317
‡1gf1	–	1	1	1-70	1	ALL
1gky†	–	2	1	1-32, 93-186	2	1-30, 81-186
			2	33-92		31-80
‡1gly	–	2	1	1-13, 245-431	2	1-20, 227-432
			2	14-244, 432-471		21-226, 441-471
‡1gmf	A	1	1	5-123	1	ALL
‡1gmp	A	1	1	1-96	1	ALL
1gox	–	1	1	1-359	1	ALL
1gp1	A	2	1	112-155	1	ALL
			2	10-111, 156-193		
‡1gpb†	–	2	1	482-831	2	485-813
			2	19-481, 832-841		19-484, 814-831
‡1gpr	–	1	1	4-161	1	ALL
‡1grc	A	1	1	1-209	1	ALL
‡1grd	A	1	1	34-114	1	ALL
‡1gst†	A	1	1	1-217	2	1-81
		2				90-217

(*continued*)

Table 1. *Continued*

A	B	C	D	E	F	G
1hbg†	–	2	1	1-71, 103-138	1	ALL
			2	72-102, 139-147		
1hc1†	–	3	1	130-294, 342-653	3	1-177
			2	5-129		178-400
			3	295-341		401-663
‡1hcc	–	1	1	1-59	1	ALL
‡1hip	–	1	1	1-85	1	ALL
‡1hiv	A	1	1	1-99	1	ALL
‡1hoe	–	1	1	1-74	1	ALL
‡1hom	–	1	1	1-68	1	1-68
‡1hrh	A	1	1	427-556	1	ALL
‡1hsa	A	2	1	1-181	2	1-175
			2	182-276		182-276
‡1hsa	B	1	1	1-99	1	ALL
‡1il1b	–	1	1	3-153	1	ALL
‡1ifc	–	1	1	1-131	1	1-131
‡1ipd	–	2	1	100-252	1	ALL
			2	1-99, 253-345		
1l53	–	2	1	1-13, 59-164	2	1-69
			2	14-58		70-164
‡1lap	–	3	1	162-484	3	162-484
			2	74-161		1-160
			3	1-73		353-484
‡1lfi†	–	4	1	435-594	4	434-595
			2	340-434, 595-691		345-433, 596-663
			3	1-91, 251-339		1-90, 252-320
			4	92-250		91-251
‡1lig	–	1	1	25-180	1	ALL
‡1lld	A	3	1	146-319	2	147-319
			2	7-77		7-146
			3	78-145		
‡1lmb	A	1	1	6-92	1	ALL
‡1lpe	–	1	1	23-166	1	ALL
‡1lts	A	1	1	4-22, 60-156	1	ALL
‡1lts	D	1	1	1-103	1	ALL
‡1mba	–	1	1	1-146	1	ALL
‡1mlp	A	1	1	1-58	1	ALL
1mnr	–	4	1	161-224	2	3-121, 344-359
			2	225-299		133-338
			3	120-160, 300-348		
			4	3-119, 349-359		
‡1ms2	A	1	1	1-96	1	ALL
‡1myg	A	1	1	1-153	1	ALL
1nsb	A	3	1	76-82, 176-265	6	108-173
			2	266-406		174-214
			3	83-175, 407-465		215-267
			4			267-314
			5			315-394
			6			395-459
‡1ofv	–	1	1	1-169	1	ALL
‡1p11	E	2	1	126-230	2	118-230
			2	15A-125, 231-244		15-117, 232-244
‡1paz	–	1	1	1-120	1	ALL
‡1pba	–	1	1	1-81	1	1-81
‡1pfk†	A	2	1	0-138, 252-304	2	0-139, 256-304
			2	139-251, 305-319		140-255, 305-319
1pgd†	–	3	1	211-253	2	33-344, 438-466
			2	181-210, 254-436		345-437
			3	1-180		
‡1pgx	–	1	1	8-77	1	ALL
1pha	–	3	1	67-128, 170-204, 241-371	2	10-101, 296-355
			2	10-66		102-295, 355-414
			3	129-169, 372-414		
1phh	–	2	1	1-45, 100-154	1	ALL
			2	46-99, 155-394		
‡1pi2	–	1	1	3-63	1	ALL
‡1pii†	–	2	1	1-255, 447-452	2	1-255
			2	256-446		256-452
‡1pk4	–	1	1	0-80	1	ALL
‡1plc	–	1	1	1-99	1	ALL
1ppn†	–	2	1	1-18, 112-207	2	1-9, 112-206
			2	19-111, 208-212		10-111
1prc	C	1	1	24-332	2	33-143, 315-332
			2			1-32, 144-314
‡1prc	H	1	1	25-82, 117-243	1	133-258
1prc	M	1	1	1-323	3	1-51
			2			52-190
			3			191-323
1pyp	–	1	1	1-280	1	ALL
‡1r69	–	1	1	1-63	1	ALL
1rat	–	2	1	1-49, 79-103	1	ALL
			2	50-78, 104-124		
‡1rbp	–	1	1	1-174	1	ALL
‡1rcb	–	1	1	1-129	1	ALL
1rhd	–	3	1	157-181, 208-274	2	159-293
			2	1-156		1-158
			3	182-207, 275-293		
‡1rn4	–	1	1	1-104	1	ALL
‡1rnb	A	1	1	2-110	1	ALL
‡1rop	A	1	1	1-56	1	ALL
1rve	A	2	1	19-47, 139-164	1	ALL
			2	2-18, 48-138, 165-245		
‡1sgt	–	2	1	16-22, 122-233	2	16-22, 129-229
			2	23-121, 234-245		23-128, 230-245
‡1snc	–	1	1	7-141	1	ALL
‡1snw	A	2	1	114-178	2	114-177
			2	179-264		178-264
‡1stp	–	1	1	13-133	1	ALL
‡1tgi	–	1	1	1-112	1	ALL
‡1tgs	I	1	1	1-56	1	ALL
1thm	–	3	1	238-279	2	1-127
			2	132-205		128-208
			3	1-131, 206-237		
‡1tie	–	1	1	1-170	1	ALL
‡1tlk	–	1	1	33-135	1	ALL
1tmd†	–	1	1	1-710	3	1-383
			2			384-494, 649-733
			3			495-648
‡1tme	1	1	1	28-256	1	31-250
‡1tnf	A	1	1	6-157	1	ALL
‡1trb	–	2	1	117-244	2	117-244
			2	1-116, 245-315		1-116, 245-316
‡1ula	–	1	1	1-289	1	ALL
‡1utg	–	1	1	1-70	1	ALL
‡1vsg	A	2	1	1-33, 88-254	2	1-32, 87-254
			2	34-87, 255-362		33-86, 255-362
1wsy†	A	2	1	173-232	2	1-188
			2	1-172, 233-265		189-268
‡1wsy†	B	2	1	9-27, 224-331	2	1-52, 86-204
			2	28-223, 332-393		53-75, 205-397
1xis	–	1	1	2-387	1	2-317
1ycc†	–	2	1	34-86	1	ALL
			2	-5-33, 87-103		
‡256b	A	1	1	1-106	1	ALL
‡2aza	A	1	1	1-129	1	ALL
2c2c†	–	2	1	34-95	3	1-62

(*continued*)

Table 1. *Continued*

A	B	C	D	E	F	G
2c2c†			2	1-33, 96-112		63-95
			3			95-112
‡2ccy	A	1	1	2-128	1	ALL
‡2cdv	–	1	1	29-107	1	ALL
‡2cpk	E	2	1	122-340	2	15-31, 126-317
			2	15-121, 341-350		33-126, 318-350
‡2ctx	–	1	1	1-71	1	ALL
‡2cyp	–	2	1	145-262	1	ALL
			2	2-144, 263-294		
2dpv	–	1	1	37-584	3	37-281, 336-410, 446-584
			2			282-335
			3			411-445
‡2fbj	H	2	1	1-118	2	1-118
			2	119-220		119-208
‡2fbj	L	2	1	1-106	2	1-104
			2	107-213		105-210
‡2fx2	–	1	1	2-148	1	ALL
‡2fxb	–	1	1	1-81	1	ALL
‡2gn5	–	1	1	1-87	1	ALL
2had	–	1	1	1-310	2	1-155, 230-310
			2			156-229
‡2hip	A	1	1	1-71	1	ALL
‡2hmq	A	1	1	1-113	1	ALL
‡2hpr	–	1	1	2-88	1	ALL
‡2liv	–	2	1	1-118, 251-325	2	1-120, 250-328
			2	119-250, 326-344		121-249, 329-344
‡2ltn	A	1	1	1-181	1	ALL
‡2mcm	–	1	1	1-112	1	ALL
‡2mev	4	1	1	13-70	1	ALL
‡2msb	B	1	1	109-221	1	ALL
2npx	–	5	1	78-115, 244-283	4	1-114
			2	1-77		115-243
			3	284-323		244-324
			4	116-243		325-446
			5	324-447		
‡2pab	A	1	1	10-123	1	ALL
‡2plv	1	1	1	6-291	1	83-202, 234-265
‡2plv	4	1	1	2-69	1	ALL
‡2pmg†	A	4	1	420-561	4	420-562
			2	1-188		1-188
			3	189-303, 388-419		189-297, 379-408
			4	304-387		298-378
‡2por	–	1	1	1-301	1	ALL
‡2reb	–	2	1	27-269	2	23-268
			2	270-328		269-328
‡2rn2	–	1	1	1-155	1	ALL
‡2rsp	A	1	1	1-124	1	ALL
‡2scp	A	1	1	1-174	1	ALL
‡2sn3	–	1	1	1-65	1	ALL
‡2stv	–	1	1	25-195	1	26-195
2tmv	P	1	1	1-154	1	ALL
‡2trx	A	1	1	1-108	1	ALL
2ts1†	–	3	1	223-319	2	248-319
			2	1-130, 173-222		1-220
			3	131-172		
2tsc	A	2	1	1-56, 146-264	1	ALL
			2	57-145		
‡2wrp	R	1	1	5-108	1	ALL
2yhx	–	3	1	19-48, 188-285, 363-458	3	20-49, 190-282, 364-432
			2	49-187		50-189, 433-451
			3	2-18, 286-362		2-19, 284-363
‡351c	–	1	1	1-82	1	ALL
‡3b5c	–	1	1	3-87	1	ALL
3bcl	–	1	1	3-358	1	ALL
‡3cd4	A	2	1	1-97	2	1-97
			2	98-178		98-178
‡3chy	–	1	1	2-129	1	ALL
3cla	–	2	1	6-91, 141-219	1	ALL
			2	92-140		
3dfr†	–	2	1	1-29, 96-162	1	ALL
			2	30-95		
‡3ebx	–	1	1	1-62	1	ALL
3enl†	–	2	1	1-33, 111-436	2	1-142
			2	34-110		143-420
‡3fgf	–	1	1	20-143	1	ALL
‡3gap	A	2	1	1-136		1-125
			2	137-208		126-208
3grs	–	1	1	18-478	3	18-57, 108-158, 293-363
			2			50-107, 159-291
			3			365-478
‡3il8	–	1	1	5-72	1	ALL
‡3mt2†	–	1	1	1-61	2	1-29
			2			30-61
‡3pgk†	–	3	1	193-235, 328-402	2	199-387
			2	236-327		1-198, 388-478
			3	1-192, 403-415		
3pgm	–	2	1	1-88, 131-230	2	1-88, 149-230
			2	89-130		89-148
3psg†	–	4	1	192-309	2	1-170
			2	125-191, 310-326		180-327
			3	10P-12		
			4	1P-9P, 13-124		
‡3rub	S	1	1	10-123	1	ALL
3sdp	A	2	1	5-74, 111-176	1	ALL
			2	75-110, 177-190		
‡3sic	I	1	1	7-113	1	ALL
4blm	A	2	1	31-86, 154-291	2	31-70, 217-291
			2	87-153		71-216
‡4bp2	–	1	1	1-123	1	ALL
‡4fd1	–	1	1	1-106	1	ALL
4gcr	–	2	1	1-83, 172-725	2	1-80
			2	84-171		83-174
‡4icb	–	1	1	0-75	1	ALL
‡4mdh	A	3	1	1-84	2	1-151
			2	85-153		152-333
			3	154-333		
‡4sbv	A	1	1	62-260	1	ALL
‡4sgb	I	1	1	1-51	1	ALL
‡4tnc	–	2	1	3-90	2	3-88
			2	91-162		101-162
5fbp†	A	2	1	6-212, 240-320	2	1-201
			2	213-239, 321-335		202-335
‡5p21	–	1	1	1-166	1	ALL
5rub†	A	5	1	393-457	2	1-139
			2	2-137, 292-316		140-457
			3	317-366		
			4	138-162, 367-392		
			5	163-291		
‡6abp	–	2	1	109-254, 286-306	2	110-253, 295-306
			2	2-108, 255-285		2-109, 254-286
‡6ebx	A	1	1	1-62	1	ALL
‡7cat	A	2	1	25-67	3	3-68
			2	68-500		69-433
			3			434-500
7tim	A	3	1	2-6, 125-230	1	ALL

(*continued*)

Table 1. *Continued*

A	B	C	D	E	F	G
7tim			2	7-61, 231-248		
			3	62-124		
8acn	–	5	1	2-15, 63-197, 271-300, 505-529	3	2-201
			2	198-270, 301-346		202-511
			3	347-504		532-754
			4	16-62		
			5	530-754		
‡8adh†	–	2	1	1-178, 318-374	2	1-175, 319-374
			2	179-317		176-318
‡8atc	A	2	1	131-291	2	144-290
			2	1-130, 292-310		1-143, 291-310
‡8atc	B	2	1	8-98	2	8-100
			2	99-153		101-153
‡8rxn	A	1	1	1-52	1	ALL

[a] Abbreviations used for column headings in this table: A, Brookhaven code; ‡ before the code indicates that the algorithm thinks that its definition is correct; † after the code indicates the definition was taken from the literature. B, chain. C, number of domains in derived definition. D, domain number. E, derived definition. F, number of domains in reference definition. G, reference definition. "ALL" indicates protein is a single domain made up of all residues. † after the name indicates the definition was taken from the literature.

has also been suggested that each of the domains can be divided into two subdomains (Kabsch et al., 1990). So residues 1-32, 70-144, and 338-375 make up subdomain Ia, and residues 33-69 make up subdomain Ib. For the second domain, residues 145-180 and 270-337 make up subdomain IIa and residues 181-269 make up subdomain IIb. DOMAK classes the protein into three domains, I, IIa, IIb with the default parameter values. If the default parameters are varied, it is possible to find all four subdomains or find only the two main domains. Thus, there is a "gray area" of domain definition where one is not sure if a subunit of the protein structure should be classed as a separate domain or whether it is merely a lobe or local compact region. By choosing a set of parameters (principally the *MSV* value), a fixed subjective limit has been set and applied objectively to the whole set.

After applying the three reliability screens described in the Materials and methods, domains from 57 proteins are found that are believed to be defined incorrectly by the algorithm. Twenty-three of the 57 proteins were incorrectly defined in comparison with the reference set. Twenty-five were from Set B. Nine definitions from Set A were picked out as incorrect.

Hence, the list of definitions automatically defined as correct is reduced to 173 (75% of the original 230 proteins). Of these,

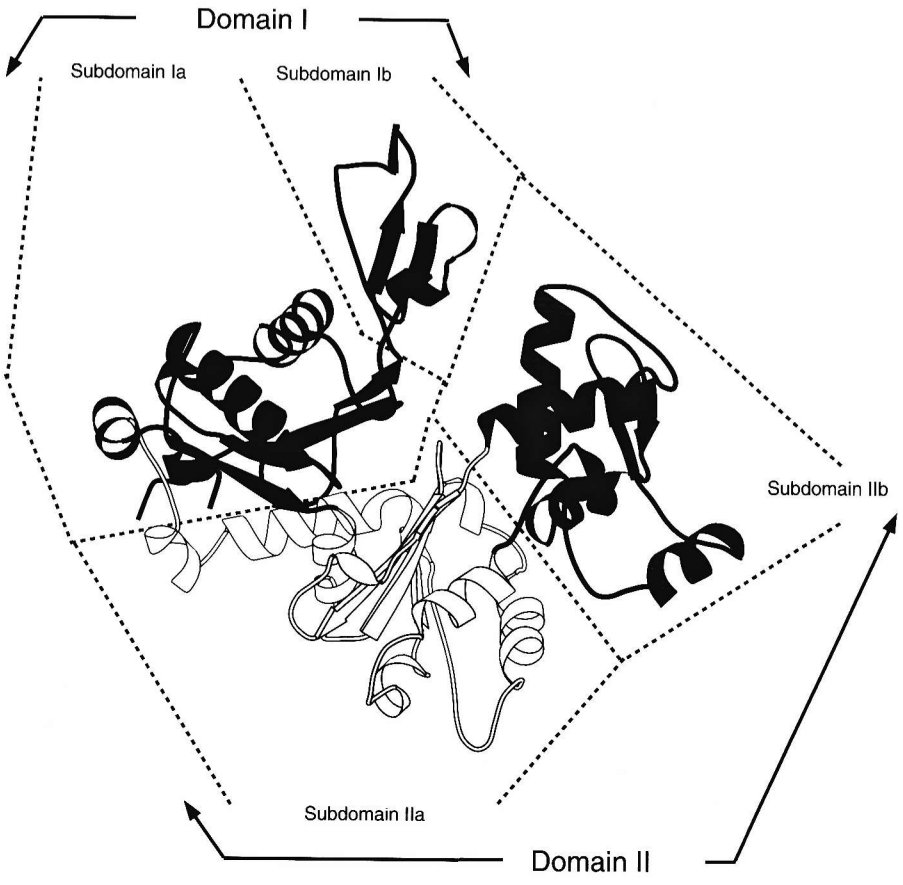

Fig. 2. Actin can be thought of consisting of two main domains, each of which can be split into two smaller subdomains. This example highlights the gray area of domain definition where one has to draw the line between what one calls a domain and what one terms a subdomain. The algorithm split this protein into three domains marked by shading (domain I, subdomain IIa, and subdomain IIb). Figure was produced using MOLSCRIPT (Kraulis, 1991).

88% match the reference set. Nine percent are from Set B and split the chain into what look like domains (Table 2). If one chooses to accept these definitions, the reliability of the algorithm rises to 97%. The five (3%) remaining structures that were incorrectly defined are listed in Table 3, together with the reasons why the algorithm gave different definitions with the default parameters. The structures that are automatically defined as correct are labeled with a ‡ in column A of Table 1.

Analysis of the derived set

The structures that the algorithm identified as correctly split can be divided on the basis of the number of domains they contain. Table 4 summarizes the number of occurrences of an *n*-domain protein. Single-domain proteins are the largest group at 75% of the set. Over the entire set there is an average of 1.3 domains per protein. The number of occurrences of an *n*-domain protein falls off rapidly as *n* is increased, and 98% of the proteins contain three or fewer domains.

Examples of a single-, two-, and four-domain proteins are shown in Figure 3 (see also Kinemage 4). Figure 3A and Kinemage 1 show trypsin (Read & James, 1988), a serine protease. It is divided into two domains, with a single cut in the middle of the chain and with both the N- and C-termini crossing back over into opposite domains, making each domain a two-segment domain, similar to the topology of the two domains shown in Figure 1C. Figure 3B illustrates the A chain of the protein phosphoglucomutase (Lin et al., 1986). It is split into four domains. The chain runs from the first domain into the first half of the second domain, passes through the third domain, comes back into the second domain to complete it, and finally makes up the fourth domain.

Figure 4 shows the distribution of the number of residues in a domain. Most domains are made up of between 50 and 100 residues. Ninety percent of the domains are comprised of less than 200 residues. The histogram tails off rapidly for large domains and there are only two domains made up of more than 400 residues (the two domains of glycogen phosphorylase).

Although the algorithm is primarily designed to search for single-segment or double-segment domains, it is possible for domains to be made up of more segments by noncontiguous "chopped segments" being added onto the domain. Table 5 summarizes the number of *n*-segment domains. A total of 81.5% of the domains found were single segment. A further 17.6% of the domains were made up of two segments. Only one three-segment and one four-segment domain were found in the final set (both the domains of glucose oxidase; Hecht et al., 1993).

The two-segment domains were subclassified on the basis of those in which there is a large difference in the relative sizes of the segments. The size of the smaller segment as a percentage of the size of the whole domain was calculated (histogram, Elec-

Table 2. *Table of domains listed as correct that have an acceptable difference to the reference definition*

Brookhaven code	Chain	What is the difference between the reference and derived definitions?
1ald	–	Reference definition is a single domain. Derived definition is acceptable.
1atn	A	See Figure 5
1atn	D	Reference definition is a single domain. This definition is acceptable.
1bbk	A	Reference definition is a single domain. This is a propeller fold structure with seven repeated units. The derived definition splits it into six domains, with one domain containing two of the repeated units.
1ezm	–	Reference definition is two single-segment domains. It is instead split into two two-segment domains.
1fnr	–	Minor difference – one derived domain is a two-segment domain.
1gal	–	Similar definition, but split into two domains not three.
1gpb	–	Minor difference – one derived domain is a two-segment domain.
1lap	–	One reference domain is split into two further parts. Two of the reference domains remain unsplit in the derived definition.
1lld	A	One reference domain is split into two further parts.
1pii	–	Minor difference.
2cpk	E	Similar definition.
2cyp	–	Reference definition is a single domain. This definition is acceptable.
3pgk	–	One reference domain is split into two further parts.
4mdh	A	One reference domain is split into two further parts.
7cat	A	Two of the reference domains remain unsplit.

Table 3. *Table of domains listed as correct that have major difference to the defined definition*

Brookhaven code	Chain	Why is there a difference?
1aai	A	Is incorrectly classed as a single domain that is made up of four segments.
1gst	A	Is incorrectly classed as a single domain.
1ipd	–	Is split into two "domain-like parts," except for the fact that a sheet is split.
1wsy	B	Both definitions split this into two domains, but the two definitions are quite dissimilar.
3mt2	–	This is a two-domain protein with each domain containing about 30 residues (smaller than the minimum domain size, MDS).

Table 4. *Number of n domain proteins*

No. of domains in protein	No. of occurrences
1	129
2	34
3	6
4	3
5	0
6	1

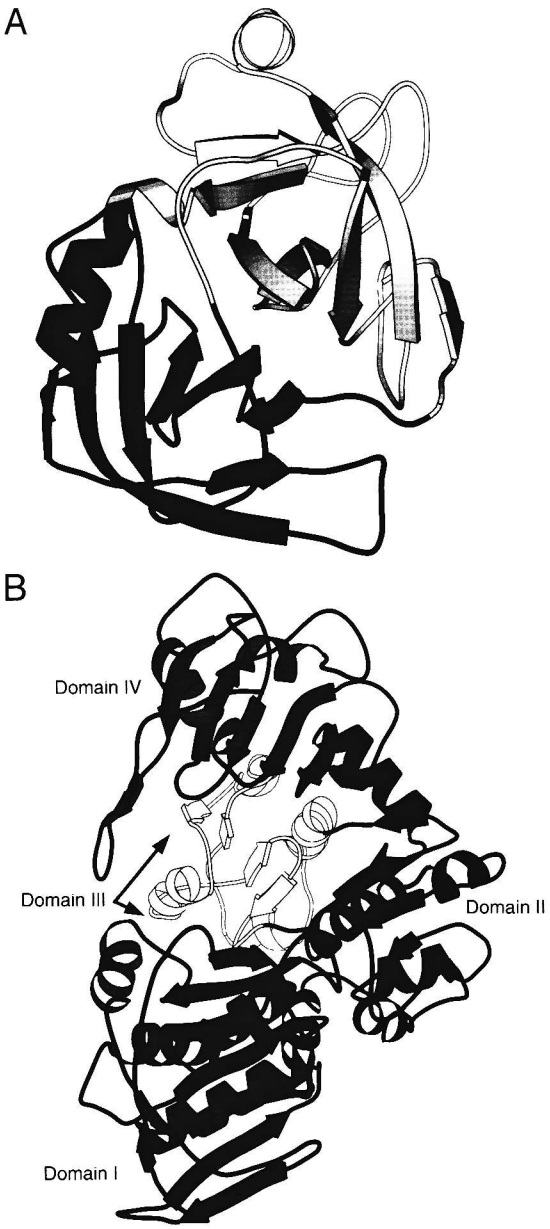

Fig. 3. A: Trypsin is classed as a two-domain protein. Topology of the chain is similar to that in Figure 1C. **B:** The *A* chain of phosphoglucomutase is split into four domains. The chain runs from domain I into the first half of domain II, into domain III, completes domain II, and finally goes into domain IV. Figures were produced using a version MOLSCRIPT (Kraulis, 1991), modified by Robert Esnouf (pers. comm.).

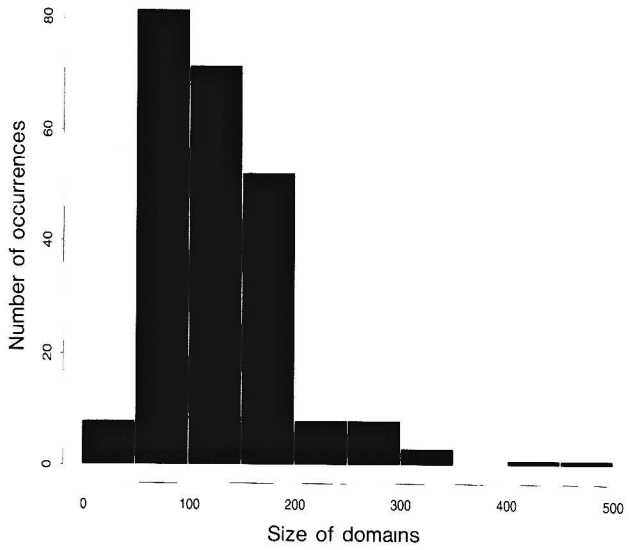

Fig. 4. Histogram showing the distribution of domain sizes.

tronic Appendix). The distribution is fairly even over the entire range, though the number of domains, in which one segment is 20–40% the size of the other, is significant.

The distance separating the residue at the end of the first segment and the residue at the start of the second segment was examined as a percentage of the size of the intervening segment. The size of the intervening segment was estimated by working out the maximum C^{α}-C^{α} separation in the domain (histogram, Electronic Appendix). The distribution appears to be normal with a peak in the range 30–40%. For 76% of the domains the separation is less than half the maximum C^{α}-C^{α} separation in the intervening segment. This shows that most inserted domains have their connections to the rest of the protein close together. A close connection may suggest that the inserted domain could be deleted without disrupting the integrity of the two-segment domain.

No correlation was found between the end-point distance and the relative sizes of the segments.

Discussion

The algorithm described in this paper can locate domains for any length of protein and is fast enough to be run routinely on the large database of protein structures. After screening, the domain definitions agree very well with conventional subjective definitions (97%). The algorithm could be developed to include the screens at an earlier stage and thus detect unlikely domains, alter the relevant constraint values, then run the analysis again.

Most of the differences between the automatically derived domain definitions and the reference definitions lie with difficulties and inconsistencies in what is meant by a "domain." The algorithm described here finds compact local regions of structure according to a set of thresholds (Table 6). However, these compact regions do not always correspond to what one would intuitively consider to be the domains in the protein. This problem is common to all previous algorithms for protein domain definition (Rossmann & Liljas, 1974; Crippen, 1978; Rose, 1979;

Table 5. *Number of n segment domains*

No. of segments in domain	No. of occurrences
1	190
2	41
3	1
4	1

Table 6. *Table of constraints*

Constraint	Subdivision	Full name	Value
MDS		Minimum domain size	40 residues
MNCC	*MNCCm*	Minimum no contact cut-off middle of chain	30 residues
	MNCCe	Minimum no contact cut-off end of chain	10 residues
MSS	*MSSm*	Minimum segment size middle of chain	25 residues
	MSSe	Minimum segment size end of chain	5 residues
sso		If percentage of secondary structure is greater than this only use secondary structure contacts	57%
MSV	*MSV*	Minimum split value	9.5
	MSVsso	Minimum split value using only secondary structure contacts	17.05
	MSVcs	Minimum split value for chopped segments	60.0
BW		β-Sheet weighting	0.1
HCD		Reduce contact density of helix to this value	10.32 contacts/residue
MDSP		Minimum size of segment for a double split	120 residues
MAC		Maximum allowed compactness	2.85 Å
ID		Increment divider	250 residues

Rashin, 1981; Wodack & Janin, 1981; Go, 1983; Zehfus & Rose, 1986; Holm & Sander, 1994; Zehfus, 1994) and is an inevitable consequence of applying an objective set of rules for domain definition to what is an essentially subjective interpretation. A major advantage of the algorithm described here is the ability to screen accurately the derived domains for domains that are unlikely to fit the normal concept of a domain. Accordingly, the final list of domains may be used with a high degree of confidence. A server of domain definitions, accessible via the World Wide Web, can be found at http://geoff.biop.ox.ac.uk.

Materials and methods

Introduction—Split value

The concept at the center of the domain identification algorithm is that residues comprising a domain make more contacts between themselves (internal contacts) than they do to the rest of the protein (external contacts). This follows from the work of Rossmann and Liljas (1974), who suggested that a domain has many short residue–residue distances within itself, but few short distances between it and the rest of the protein. Thus, the ratio of the number of internal contacts to the number of external contacts should be large for a domain. Two residues are defined to make a contact if a heavy atom in one residue is within 5 Å of a heavy atom in the other.

If the protein is split into two arbitrarily chosen parts, A and B, then the quantity

$$(int_A/ext_{AB}) * (int_B/ext_{AB})$$

can be calculated, where int_A is the number of internal contacts in A, int_B the number of internal contacts in B, and ext_{AB} the number of contacts between A and B. This quantity is referred to as the split value. The split value will be large if the A and B are distinct. If the two parts are not distinct (i.e., correlated), then the split value will be small.

A simple implementation of the idea

Consider chopping the protein chain into two parts of segments between residues i and $(i + 1)$. A segment can consist of any number of residues, but the residues must form a continuous sequence along the chain. Segment A then consists of residues 1 to i and segment B of residues $(i + 1)$ to N, where N is the number of residues in the chain. The split value can then be calculated for $1 \leq i < N$. Figure 5 illustrates a graph of split value against i for the T-cell surface glycoprotein, CD4 (Ryu et al., 1990). The split value has a large peak at $i = 97$, indicating that the protein should be split into two domains at this point. Once split, the two domains can themselves be individually scanned to find the maximum split values and hence the best positions to split them into new domains, which again can be scanned and split and so on. By placing a limit on the minimum number of residues in a domain (minimum domain size, *MDS*) and/or defining a minimum split value (*MSV*) below which the two parts are considered to be correlated and not divisible into smaller domains, the process of division can be stopped. The result is a series of "cuts" defining how the chain should be split into separate domains.

Allowing for two-segment domains

Consider a domain made up of a single segment that consists of residues k to l, inclusive, which is scanned to find further domains.

Method 1—A single-segment scan (Fig. 6A)

Segment A is chosen by cutting the chain at two points, x and y. Therefore, B can consist of up to two segments, B_1 and B_2, depending on the positions of the boundaries. The split value is calculated for all possible segment A's formed by varying x and y. The maximum split value is stored, together with the corresponding values of x and y, called x^{max} and y^{max}, which define A^{max}. The maximum split value is compared with *MSV* and if A^{max} is not correlated with B^{max}, then A^{max} can be "extracted" from the "parent" domain to form a new "child" do-

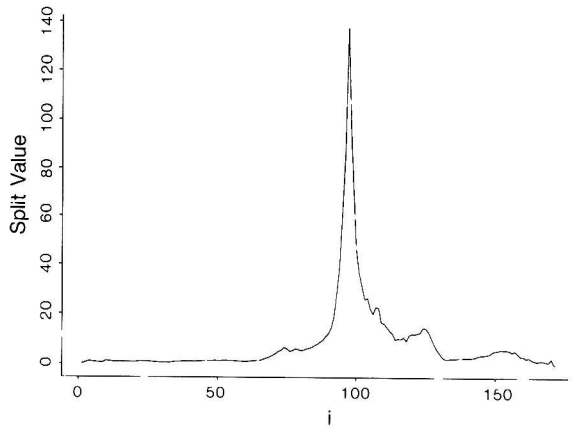

Fig. 5. Graph showing how the split value varies with i for 1CD4. The protein is cut into two segments, A and B, between residues i and $(i+1)$. The graph shows a large peak at $i = 97$, indicating that the protein should be split into two domains at this point. Although this example is a relatively easy case of a two-domain protein, it illustrates the basic method well. The fact that there are two clear domains is reflected by the size and narrowness of the peak.

main (also referred to as a "subdomain"). The treatment of B^{max} is the same for all three scans and is shown at the end of Method 3.

Note that the single segment scan would be able to deal with both the situations that arise in Figure 1A and B. However, it would not be able to deal with the case shown in Figure 1C. To allow for this eventuality, a two-segment scan is used.

Method 2—A two-segment scan (Fig. 6B)

A is made up of two segments, A_1 and A_2, formed by cutting the chain at four points, x_1, y_1, x_2, and y_2. The split value between A_1 and A_2 must show them to be correlated when compared with MSV. B can be made of up to three segments, depending on the positions of the boundaries. The split value is calculated for all possible segment A's formed by varying x_1, y_1, x_2, and y_2. The maximum split value is stored, together with the corresponding values of x_1, y_1, x_2, and y_2, called x_1^{max}, y_1^{max}, x_2^{max}, and y_2^{max}, which define A^{max} and B^{max}. The maximum split value is compared with MSV and if A^{max} and B^{max} are not correlated, the parent domain is split at this point. A^{max} goes on to form a two-segment child domain.

Method 3—A two-segment scan of a two-segment domain (Fig. 6C)

Now consider a domain made up of two segments, consisting of residues k_1-l_1 and k_2-l_2. The algorithm scans this domain for subdomains in the following way.

A is made up of two segments A_1 and A_2, formed by placing four boundaries at x_1, y_1, x_2, and y_2. Note that one of the boundaries of both components of segment A must lie on the boundary of the parent domain. The split value between A_1 and A_2 must show them to be correlated when compared with MSV. B can consist of up to two segments. This split value is calculated for all possible segment As formed by varying x_1, y_1, x_2, and y_2. The maximum split value is stored, together with the corresponding values of x_1, y_1, x_2, and y_2, called x_1^{max}, y_1^{max}, x_2^{max}, and y_2^{max}, which define A^{max} and B^{max}. The maximum split value is compared with MSV and if A^{max} and B^{max} are not correlated, the parent domain is split at this point. A^{max} goes on to form a two-segment child domain.

For all three scans, if B^{max} consists of only one segment, it is considered to form a single-segment child domain. If B^{max} consists of two segments, the split value between these two parts is calculated. If the two segments are correlated, they are placed together to form a single child domain made up of two segments, otherwise, they are considered to be two separate, child domains. If B^{max} consists of three segments, the split values between all pairs are calculated. If none of the pairs are correlated, the segments are considered to form three distinct child domains. If one of the pairs is correlated, the two segments are placed together to form a two-segment child domain, the leftover segment forming a single-segment child domain on its own. If two or three of the pairs are correlated, the pair with the highest degree of correlation (i.e., lowest split value) is placed together to form a two-segment child domain, the leftover segment again forming a single-segment child domain on its own.

Applying the methods to divide a protein

Armed with these methods, the algorithm will start off treating the chain as a single-segment domain and divide it using Method 1 or 2, whichever yields the higher split value. If a two-segment domain is found at any point, it is scanned using

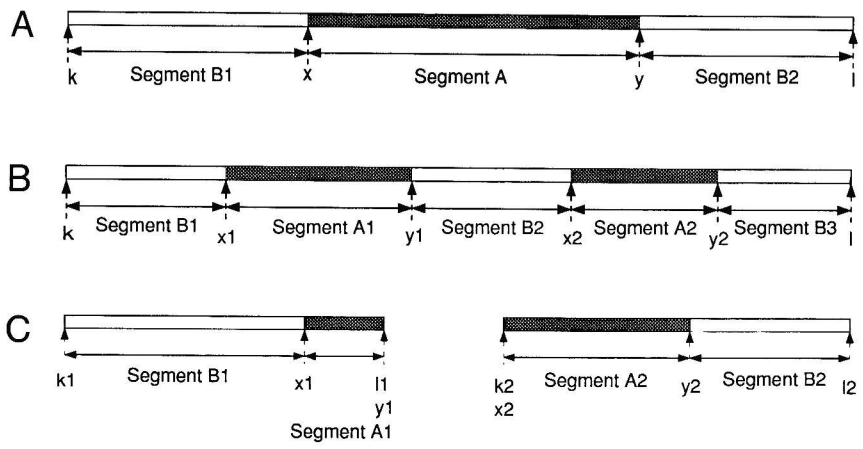

Fig. 6. A: Single-segment scan in which A is made from a single segment extracted from the parent domain, splitting it into two parts. **B:** Two-segment scan in which A is made from two segments, splitting the parent domain into three. **C:** Two-segment scan of a two-segment parent domain. A is made up of two segments, one in each of the parent segments. Note that, in this case, one end of each segment must be at the end of a parent segment.

Method 3. The algorithm continues to divide the protein, until it is checked by one of the constraints. Constraints are described in the next section and are also present to allow the algorithm to be flexible and fast.

Note that none of these methods will deal with domains consisting of three or more segments. Domains such as these are not dealt with explicitly in the algorithm at the scanning stage because the complexity of the scan would increase rapidly. However, they are allowed implicitly at a later stage (described below). Such domains are found to be quite rare, making up only a small fraction of the total number of domains in the database.

Additional details

The *MSV* is used to decide whether two segments are distinct or correlated. If the split value found is less than the *MSV*, the two segments are correlated, otherwise they are distinct.

A segment can consist of any number of residues, but the residues must form a continuous sequence along the chain. There are three types of constraints on the number of residues in a segment (Table 6): minimum domain size (*MDS*), minimum no contact cut-off (*MNCC*), and minimum segment size (*MSS*). They are chosen such that $MDS > MNCC > MSS$. A segment that has $size \geq MDS$ and is distinct from the rest of the parent domain is considered to form a child domain. This constraint provides control over the minimum size of the domain and prevents the protein being split into small pieces. A segment with $size < MDS$ but $\geq MNCC$, that is found to be distinct from the rest of the parent domain, is not large enough to form a child domain. Instead, it is classed as a "chopped segment." Chopped segments allow the algorithm to remove small segments from a domain that are not strongly correlated to it and later reassign them to other domains, or back to the original one. This allows domains to consist of more than two noncontiguous segments. The treatment of chopped segments is discussed below. Segments with $size < MNCC$ but $\geq MSS$, are used by two-segment scans (both Methods 2 and 3). In these scans, two segments can come together to form a single domain. It is possible that one of the segments may be small. To allow for this, segments that have a size in this range are only allowed if they are correlated with another segment, such that the total size of the two segments is $\geq MDS$. Segments with $size < MSS$ are not allowed, thus preventing very small segments from occurring. When domains are inspected, one often finds small segments at the N- or C-termini that cross domains. Segments in the middle of the chain as small as this do not cross domains. Thus, to allow for this difference, *MNCC* and *MSS* are divided into two categories: segments that are present in the middle of the chain and those that have one end connected to the end of the chain, to give *MNCCm*, *MNCCe*, *MSSm*, and *MSSe*. The values of these constraints are given in Table 6.

Helices form a relatively large number of contacts per residue (contact density) when compared to coil and β-sheet. The average contact density in 2,446 coil regions, 1,324 helices, and 1,563 β-strands was found to be: 24 ± 7 contacts/residue, for helices, 10.3 ± 6.6 contacts/residue for coil, and 3.3 ± 2.2 contacts/residue for strands. Accordingly, helical regions have a tendency not to be split, but more importantly, they raise the number of internal contacts in the segment that contains them. This can lead to segments containing helices being split incorrectly. To compensate for this, the number of internal contacts in a helix-containing segment is reduced to the average level for coil regions. The value to which it is reduced is termed helix coil density (*HCD*).

β-Sheets may sometimes be split across domains. A constant *BW* (standing for β-sheet weighting) is used to reduce the likelihood of this occurring. The number of external contacts between two regions is increased by *BW* percent for every hydrogen bond (as defined by DSSP [Kabsch & Sander, 1983]) between strands that spans the two regions. Therefore, the greater the number of strand-forming hydrogen bonds that bridge two regions, the less likely they are to be distinct.

Once all the domains have been found, their compactness is checked. If a domain is found to be noncompact, it is combined with the domain with which it has the lowest split value. The process is repeated until either all the domains are compact or all the domains have been combined together. A domain is defined as noncompact if its radius of gyration deviates from a theoretical curve (of radius of gyration against size of the domain) by more than the constraint maximum allowed compactness (*MAC*) (Russell, 1993).

Increasing the speed of execution

The speed of the domain scan depends on the size of the segment being analyzed. If the segment contains N residues, there are N places at which a boundary may be placed. A Method 1 scan cuts the segment twice and so there are $N^2/2$ possible splits. A Method 3 scan has restrictions on where segments may start and end. Suppose it contains two segments of size N_1 and N_2. Each segment is effectively scanned twice by a single cut. Therefore, the speed of the scan can be given by $(2N_1)(2N_2) = 4N_1N_2$. A Method 2 scan splits the domain four times and hence its speed varies approximately as $N^4/4$. Restriction on segment sizes helps reduce the number of combinations, but the algorithm's speed can be increased further in the following ways.

Small segments are unlikely to contain two-segment domains. Therefore, a lower bound is placed on the minimum number of residues in a segment, minimum double split (*MDSP*). Any single-segment domains with $size < MDSP$ are assumed to contain single-segment domains only. This prevents the algorithm from performing a two-segment scan on the segment and thus saves time.

If the percentage of secondary structure (i.e., helix and strand) in the domain being scanned is greater than the value given by *sso*, the algorithm uses only those contacts to and from secondary structure elements. Secondary structure element definitions are taken from the program DSSP (Kabsch & Sander, 1983), using "H" for helix and "E" for strand.

It was found that, in cases where only the secondary structure was used, the maximum split values were generally higher than they would be had the same domains been scanned using all contacts. To take the difference in maximum split values into account, the cases in which only secondary structure contacts are being used are compared against the variable *MSVsso*.

Although the above restrictions cut down on the number of combinations, once the actual number of residues being used rises above 250–300, the algorithm still takes an unreasonably long time to execute. To circumvent this problem, some "pruning" of the search tree is done on Method 2 scans (see Fig. 6B). The assumption is made that, if two segments are correlated,

increasing the size of one segment will not make the segments distinct.

Although tree-pruning is successful in most cases, it is not able to speed up others. In order to speed up all scans, the split value is not calculated at every position for large segments. Instead, the position of the cutting boundaries is moved by an "increment," skipping over intervening residues. The increment is calculated by dividing the size of the segment being analyzed by the increment divider (*ID*) and adding one. Note that this results in the cuts corresponding to the *MSV* being over a range of residues rather than at specific residues. The same situation occurs when only secondary structure is considered because the split value will remain unchanged as the cut boundary passes over nonsecondary structure residues. In both these cases, once the range of the cut boundaries is known, the algorithm goes back and calculates all the split values for all residues in the range using all contacts. The combination that gives the highest split value is the one used. Using these methods, analysis time was reduced from 11 h to 1 min on the three-domain protein BirA (Wilson et al., 1992) (for a Silicon Graphics Indy R4000 PC).

Screening the results

To be useful, any automatic algorithm must be able to tell when the definitions it has produced are likely to disagree with the expected standard. Three rules about domains were derived to enable the algorithm to identify such examples.

1. *Count the number of segments in a single-domain protein.* Single-domain proteins may have chopped segments removed that are later reassigned or may be split into domains that are recombined on the basis of compactness. If the number of segments that the final domain was split into is large, then the domain is unlikely to be a true single-domain protein. Single-domain proteins made up of four or more segments were flagged for further visual inspection (table, Electronic Appendix).

2. *Calculate the number of residues per segment for domains consisting of two or more segments.* If this is small, it is unlikely that the domain is a real domain. This suggest a lower limit on the size of such domains, which is larger than *MDS*. The limit chosen was 50 residues per segment (table, Electronic Appendix).

3. *For a single-segment domain inserted into a domain of two or more segments, calculate the ratio of the size of the domain into which the inserted domain is placed to the size of the inserted domain.* If the ratio is large, the inserted domain is unlikely to be a real domain. The limit set was 1.6 (table, Electronic Appendix).

Implementation

The algorithm was implemented as a program written in ANSI C called DOMAK ("DOMain MAKer"). All the times are for a Silicon Graphics Indy R4000 PC (32 MByte memory, no secondary cache). The program requires output files from the programs DSSP (Kabsch & Sander, 1983) and CONTACTS (R.B. Russell, pers. comm.) and also the Brookhaven file. CONTACTS is a program that calculates all heavy atom contacts in a protein. The output from DOMAK shows the steps taken to find the final list of domains, which are listed in STAMP (Russell & Barton, 1992) format. An input file for RASMOL (R. Sayle, 1992, RASMOL, molecular visualisation program, e-mail: rasmol@ggr.co.uk) to display the domains found is also produced. Further details are given in the DOMAK user guide (A.S. Siddiqui, 1994).

Reference domain definitions

A set of 275 nonredundant protein structures was derived from the Brookhaven database. The nonredundancy is based on sequence rather than structure, so some structures from the same family appear in the set. The structures were examined by Dr. R.B. Russell (pers. comm.) and subjectively split into domains using knowledge of protein folds and on the basis that domains are globular units that are distinct from the rest of the structure. For proteins in this set that contained more than one domain, the literature was searched for domain definitions in the original publications that described the crystal structure. This set is referred to as the reference set, as shown in Table 1. Table 1 also shows which definitions were derived from the literature (identified by a † after the name).

It was not possible to produce DSSP (Kabsch & Sander, 1983) files for 40 of the structures. CONTACTS could not be run on a further four structures because it requires all atoms to be present in the file. DOMAK, in its current form, has not been designed to deal with structures in which domains are made up of more than one chain. Therefore, kallikrein A was excluded from the set. However, it is conceptually simple to extend DOMAK to handle this case. The final set of protein structures analyzed was 230. DOMAK required 16.5 h of CPU to complete the analysis on this set, giving an average time of 4.3 min per protein. Calculation of contacts requires less than 2 min for the largest proteins (glycogen phosphorylase, 823 residues, took 101 s) and just over 1 s for the smaller ones (metallothionein isoform II, 62 residues; 1 s).

Optimization of parameters

There are 14 independent DOMAK parameters (Table 6) for which suitable values had to be determined. Constraints on segment sizes (*MSS*) were derived by taking the smallest values of these constraints that appear in the set of domains that was derived by eye. *MDS* was chosen by looking at the sizes of domains in the same set. The smallest domain size in this set is actually 30, but this is exceptional so a size of 40 residues was chosen. *MNCC* values have not been optimized. *sso* was chosen to provide a compromise between speed and accuracy. If the amount of secondary structure in the segment is small, looking at secondary structure contacts only will not be accurate enough. The value was chosen by looking at two examples in which a sheet was being split (Brookhaven codes 1PHA and 1IPD). *HCD* was set simply to the value of the average contact density in coil regions. *MAC* was derived by looking at the compactness of the domains that had been split by eye and choosing a value that encompassed most of them. *MSV*, *MSVsso*, and *MSVcs* were derived by looking at the behavior of five examples because these values were altered (1BBK [A chain], 1AMA, 1RHD, 1ALD, 1PHH). As with any analysis that categorizes proteins on the basis of cut-off values, there are compromises made in choosing the cut-off values. The values found produce good results

over the entire set, however, it may be possible to optimize them further.

Supplementary material in the Electronic Appendix

Subdirectory Siddiqui.SUP in the Electronic Appendix contains three tables showing the proteins that are filtered out by the three screens. It also contains two histograms as Postscript files. One shows the distribution of the size of the smaller segment of a two-segment domain as a percentage of the size of the domain. The other shows the distribution of the distance separating the ends of an inserted segment as a percentage of its size.

Acknowledgments

We thank Dr R.B. Russell for providing the program CONTACTS and the database of domains. We thank Professor L.N. Johnson for support. A.S. is funded by a Biotechnology and Biological Sciences Research Council studentship and is a member of Worcester College, Oxford. G.J.B. thanks the Royal Society for support.

References

Baron M, Campbell ID. 1991. Protein modules. *Trends Biochem Sci 16*:13-17.

Bowie JU, Eisenberg D. 1993. Inverted protein structure prediction. *Curr Opin Struct Biol 3*:437-444.

Bryant SH, Lawrence CE. 1993. An empirical energy function for threading protein sequence through the folding motif. *Proteins Struct Funct Genet 16*:92-112.

Campbell ID, Baron M. 1991. The structure and function of protein modules. *Philos Trans R Soc Lond (Biol) 332*:165-170.

Crippen GM. 1978. The tree structural organisation of proteins. *J Mol Biol 126*:315-332.

Go M. 1983. Modular structural units, exons and function in chicken lysozyme. *Proc Natl Acad Sci USA 80*:1964-1968.

Hecht HJ, Kalisz HM, Hendle J, Schmid RD, Schomburg D. 1993. Crystal structure of glucose oxidase from *Aspergillus niger* refined at 23 Å resolution. *J Mol Biol 229*:153-172.

Holm L, Sander C. 1994. Parser for protein folding units. *Proteins Struct Funct Genet 19*:256-268.

Jones DT, Taylor WR, Thornton JM. 1992. A new approach in protein fold recognition. *Nature 358*:86-89.

Kabsch W, Mannherz HG, Suck D, Pai EF, Holmes KC. 1990. Atomic structure of the actin:dnase I complex. *Nature 347*:37-44.

Kabsch W, Sander C. 1983. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers 22*:2577-2637.

Kraulis P. 1991. MOLSCRIPT: A program to produce both detailed and schematic plots of protein structure. *J Appl Crystallogr 24*:946-950.

Lin Z, Konno M, Abad-Zapatero C, Wierenga R, Murthy M. RN, Ray WJ, Rossmann MG. 1986. The structure of rabbit muscle phosphoglucomutase at intermediate resolution. *J Biol Chem 261*:264-274.

Nishikawa K, Ooi T. 1972. Tertiary structure of proteins II. Freedom of dihedral angles and energy calculations. *J Phys Soc Jp 32*:1338-1347.

Nishikawa K, Ooi T, Isogai Y, Saito N. 1972. Tertiary structure of proteins I. Representation and computation of the conformation. *J Phys Soc Jpn 32*:1331-1337.

Patthy L. 1994. Introns and exons. *Curr Opin Struct Biol 4*:383-392.

Phillips DC. 1970. *British biochemistry, past and present*. London: Academic Press. pp 11-28.

Rashin AA. 1981. Location of domains in globular proteins. *Nature 291*:85-86.

Read RJ, James MN. 1988. Refined crystal structure of *Streptomyces griseus* trypsin at 17 Å resolution. *J Mol Biol 200*:523-551.

Richardson JS. 1981. The anatomy and taxonomy of protein structure. *Adv Protein Chem 34*:246-253.

Rose GD. 1979. Hierarchic organisation of domains in globular proteins. *J Mol Biol 134*:447-470.

Rossmann MG, Liljas A. 1974. Recognition of structural domains in globular proteins. *J Mol Biol 85*:177-181.

Russell RB. 1993. Computer analysis of protein sequence and structure [thesis]. Oxford, UK: University of Oxford.

Russell RB. 1994. Domain insertion. *Protein Eng 7*:1407-1411.

Russell RB, Barton GJ. 1992. Multiple protein sequence alignment from tertiary structure comparison: Assignment of global and residue confidence levels. *Proteins Struct Funct Genet 14*:309-323.

Ryu S, Ryu SE, Kwong PD, Truneh A, Porter TG, Arthos J, Rosenberg M, Dai XP, Xuong NH, Axel R, Sweet RW, Hendrickson WA. 1990. Crystal structure of an HIV-binding recombinant fragment of human CD4. *Nature 348*:419-426.

Wetlaufer DB. 1973. Nucleation, rapid folding, and globular intrachain regions in proteins. *Proc Natl Acad Sci USA 70*:697-701.

Wilson KP, Shewchuk LM, Brennan RG, Otsuka AJ, Matthews BW. 1992. The *E. coli* biotin holoenzyme synthetase/biorepressor crystal structure delineates the biotin and DNA-binding domains. *Proc Natl Acad Sci USA 89*:9257-9261.

Wodak SJ, Janin J. 1981. Location of structural domains in proteins. *Biochemistry 20*:6544-6552.

Wodak SJ, Rooman MJ. 1993. Generating and testing protein folds. *Curr Opin Struct Biol 3*:247-259.

Zehfus MH. 1994. Binary discontinuous compact protein domains. *Protein Eng 7*:335-340.

Zehfus MH, Rose GD. 1986. Compact units in proteins. *Biochemistry 25*: 5759-5765.