doi:10.1016/S0022-2836(03)00269-9 J. Mol. Biol. (2003) 328, 749–767

JMB

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Exhaustive Enumeration of Protein Domain Families

Andreas Heger* and Liisa Holm

EMBL-EBI, Wellcome Trust Genome Campus, Hinxton Cambridge CB10 1SD, UK

Domains are considered as the basic units of protein folding, evolution, and function. Decomposing each protein into modular domains is thus a basic prerequisite for accurate functional classification of biological molecules. Here, we present ADDA, an automatic algorithm for domain decomposition and clustering of all protein domain families. We use alignments derived from an all-on-all sequence comparison to define domains within protein sequences based on a global maximum likelihood model. In all, 90% of domain boundaries are predicted within 10% of domain size when compared with the manual domain definitions given in the SCOP database. A representative database of 249,264 protein sequences were decomposed into 450,462 domains. These domains were clustered on the basis of sequence similarities into 33,879 domain families containing at least two members with less than 40% sequence identity. Validation against family definitions in the manually curated databases SCOP and PFAM indicates almost perfect unification of various large domain families while contamination by unrelated sequences remains at a low level. The global survey of protein-domain space by ADDA confirms that most large and universal domain families are already described in PFAM and/or SMART. However, a survey of the complete set of mobile modules leads to the identification of 1479 new interesting domain families which shuffle around in multi-domain proteins. The data are publicly available at ftp://ftp.ebi.ac.uk/pub/contrib/heger/adda.

Keywords: homology; evolution; protein family; domain; maximum likelihood

*Corresponding author

Introduction

Complexity in biology has evolved through modification and recombination of existing building blocks instead of invention from scratch. In the protein world these building blocks have been termed "domains"[1,2] and the identification and characterisation of new domains and domain families is a major goal of protein science.

Grouping domains into families is useful in two ways. Firstly, it leads to more sensitive detection of new members and improved discrimination against spurious hits. The essential conserved features in a family are manually expressed by profiles[3] (position-specific scoring matrices) or hidden Markov models[4] or patterns[5] (regular expressions). Secondly, having established family membership, a query sequence can be placed in the context of the evolutionary tree of the family for accurate functional inference. It is also easier to spot inconsistent similarity-derived annotations in the context of an evolutionary tree.

Traditionally, domain families have been defined manually. Recently, automated methods have been developed that systematically try to find shared building blocks between proteins. The most sensitive methods employ exhaustive structural comparisons, but are limited by the availability of structural data, which are still scarce.[6,7] More complete methods in terms of protein space coverage use sequence data alone. Here, we present an exhaustive sequence-based domain decomposition and family classification.

Protein domain family classification can be considered as a graph partitioning problem. All-against-all pairwise comparison of protein sequences yields a view of the geometry of protein

Present address: A. Heger, Institute of Biotechnology, University of Helsinki, P.O. Box 56, 00014, Finland.

Abbreviations used: PDB, Protein Data Bank; SCOP, structural classification of proteins; PFAM, protein families database; SMART, simple modular architecture research tool.

E-mail address of the corresponding author: andreas.heger@helsinki.fi

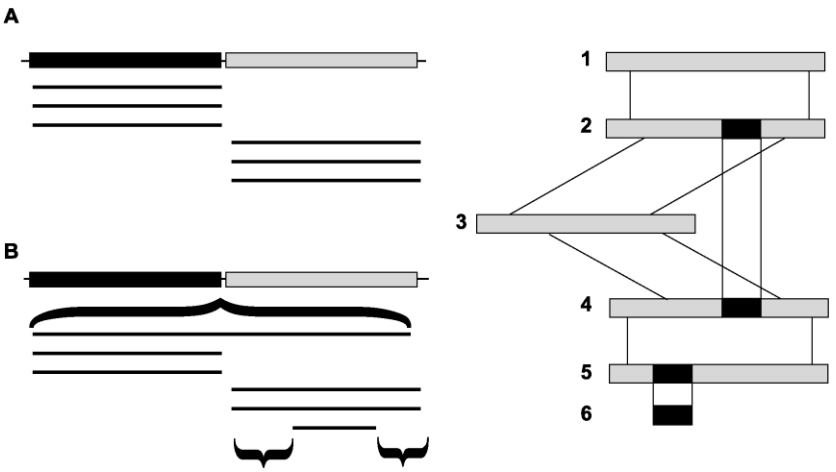

Figure 1. Key concepts in ADDA. Left: block structure of multiple alignments. A, The ideal case of a query sequence of two domains with the local alignments to its neighbours. In the ideal setting the multiple alignment exhibits a block structure, where the domain structure of the query sequence is immediately obvious. B, The real situation. Alignments between multi-domain proteins have to be split (upper bracket). At the same time, alignments to a motif or fragment do not cover all residues in a domain (lower brackets). Right: a global view corrects for motifs and fragments. Six sequences (horizontal bars) are shown with alignments between them (thin lines). Sequence pair 2,4 only aligns in a short conserved motif. Linking sequence 3 and sequences 1 and 5 from sub-families indicate that the domain is larger than the motif. Sequence 6 is a fragment, but the truncated alignment is compensated for by the alignment between sequences 4 and 5.

sequence space. Neighbour lists of each sequence induce a representation of sequence space as a graph with vertices and edges. In the sequence space graph, sequences are vertices and alignments between sequences are edges. The weight of an edge connecting two sequences represents their degree of similarity given an appropriate measure. Dense clusters in this sequence space graph correspond to families of related proteins.[8,9]

At biologically interesting levels of similarity the majority of sequences in the sequence space graph belong to a single huge connected component due to spurious similarities and multi-domain proteins. Therefore, the central challenges of protein family classification are to split sequences into domains and remove spurious links between non-homologous domains. As a result, the graph is decomposed into smaller clusters of biological relevance, i.e. domain families.

Methods for partitioning the sequence space graph are an area of active research.[10] Many approaches decompose the graph based on edge weight,[9] graph topology,[11] or edge weight and density.[12] These approaches do not split sequences into domains so that multi-domain proteins can pull unrelated domains into each other's neighbourhood. Of methods that address the domain decomposition problem, PRODOM[13] splits sequences into domains based on the greedy assumption that the shortest sequence and aligned sub-sequences always correspond to full-length domains, while DOMO[14] maps sequence termini onto multi-domain proteins. Both of these methods assume a clean input data set devoid of fragments and other artefacts.

With ADDA we explicitly model the noise in the sequence databases using a "block model" of multiple alignments. The block model incorporates noise due to sequence fragments and either truncated or spurious alignments. A global optimisation involving all sequences ensures that domain boundaries are placed consistently. After domain decomposition, domains are clustered into families based on sequence similarity.

Domain decomposition

In an ideal world, alignments covered domains completely and no two proteins shared the same domain combination in the same order. In this ideal world, a multiple alignment built from a sequence database search with a multi-domain protein exhibited a block structure (Figure 1, left) as a result of its domain composition. In the real world, the block structure is confused by three types of noise. (1) Multi-domain proteins. Aligning adjacent domains in two protein sequences results in a single alignment. In this case, one alignment represents the recurrence of more than one domain and thus is longer than a single domain and has to be split. (2) Motifs and fragments. Local alignments tend to be truncated if the sequences are distant homologs. Here, one alignment represents the recurrence of a partial domain resulting in residues not covered by the alignment. Similarly, fragments cause alignments to end before domain boundaries. (3) Spurious alignments. Non-homologous regions can be aligned, sometimes giving significant scores. The alignments might match anywhere on the sequence and thus give misleading information about domain length or location.

ADDA models noise due to multi-domain proteins, motif alignments, fragments, and spurious links. It defines an objective function that quantifies the deviation from the ideal block structure for a given partition of sequences into domains.

The objective function is optimised globally, i.e. simultaneously for all proteins in the sequence set.

Table 1. The sequence space graph decomposition by ADDA

			Families		Largest cluster	
	Sequences	Domains	Non-singletons	Singletons	Sequences	Domains
nrdb40	249,264	450,462	33,879	168,548	3267	4803
nrdb	782,238	1,367,789	79,965	122,462	32,673	34,054

The global view allows us to identify joined alignments due to multi-domain proteins and truncated alignments due to motifs and fragments (Figure 1 (right)). The optimisation step includes evidence from all sequences and can thus balance between cutting too little, i.e. unresolved multi-domain proteins, and cutting too much, i.e. fragmented sequences due to cutting at every alignment end.

Clustering

After splitting sequences into domains, the domains are clustered into families. We assume that protein sequences of a given family fluctuate around a stable point in sequence space given constant evolutionary constraints ("punctuated equilibria"[15]). If the latter change, for example, if an enzyme starts working on a new substrate, new variants derived from the family will move to a new location in sequence space: a new sub-family has been created. Consecutive changes leave a foot-print in sequence space that allows walking from any sub-family to any other either directly, if similarity is within the detection range of sequence profile models, or *via* a sequence of intermediate steps.

With ADDA, we follow this foot-print of a protein domain family in sequence space. Evolutionarily related domains are assumed to occupy continuous neighbourhoods. Unrelated domain families should be demarcated by a sharp boundary with dissimilar sequence patterns on either side. Unification proceeds by domain walking between closest neighbours where each step is checked by pairwise profile–profile comparison between the adjacent domains. Rejected steps result in domain family boundaries.

Outline

Here we present the domain decomposition of the complete sequence space graph using ADDA. We systematically survey the set of domain families and present a set of 1476 interesting domain families absent from the major manually curated domain databases. We conclude with a rigorous validation of the method.

Results

Overview of the protein universe

Here, we partition a graph of 782,238 non-identical sequences (*nrdb*). Firstly, redundant sequences are removed at 40% identity.[16] The resulting graph contains 249,264 vertices (*nrdb40*) and 25 Mio edges. In the first stage, the graph is partitioned into 450,462 domains (Table 1). In the second stage, these domains are assigned to 33,879 protein domain families containing more than one member and 168,548 singletons. Singletons are mostly due to sequence masking: 102,953 of all singleton domains contain at least one masked region of at least five residues leaving 65,615 true singletons.

Mapping the domains back onto *nrdb* yields 1,367,789 domains in 79,965 domain families and

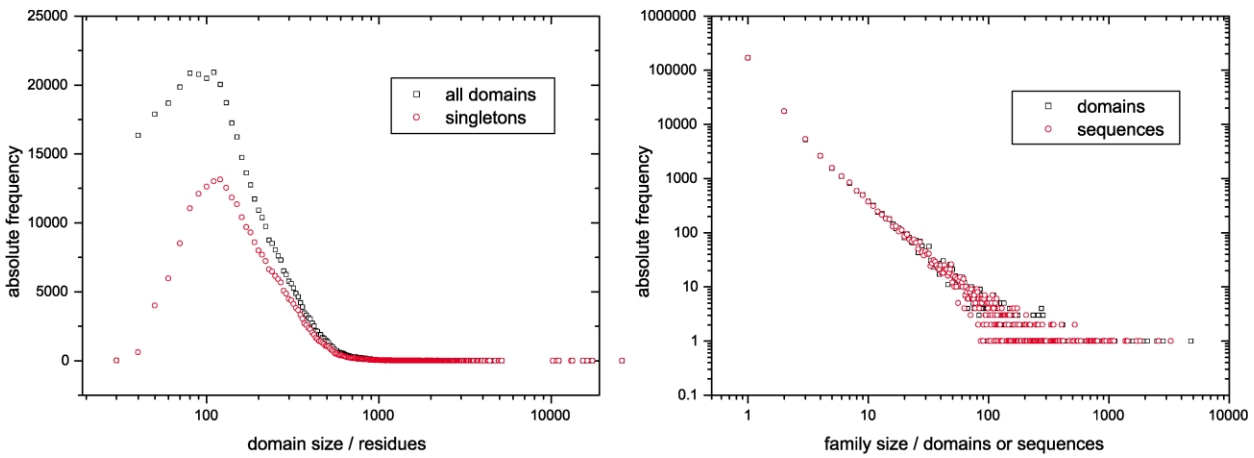

Figure 2. Cluster characteristics. Left: distribution of domain size. Bins are labelled by maximum value. Right: family size distribution.

122,462 singletons. Below, all results are reported for the *nrdb40* graph, as it corrects for bias in *nrdb*.

Both the distribution of domain and family size follow typical distributions (Figure 2). Domain sizes peak at around 100 residues. The absence of a peak at smaller lengths demonstrates that ADDA avoids excessive fragmentation. Singleton domains tend to be shorter, as many inter-domain linkers fall into this category. The distribution of family size is linear in a log–log plot. There are few domain families with many members, but many domain families with few members.

Example: homeobox domains

The homeobox domain is a DNA-binding domain in *Drosophila* and other animals. Proteins sharing this domain are prominent in cell development. ADDA locates the domain perfectly and assigns it to two major clusters, one containing 82%, and the other 13% of all homeobox domains found by PFAM. The domain is found frequently associated with other domains (Figure 3). The domain decomposition of various sequences reveals the modularity of the sequences similar to

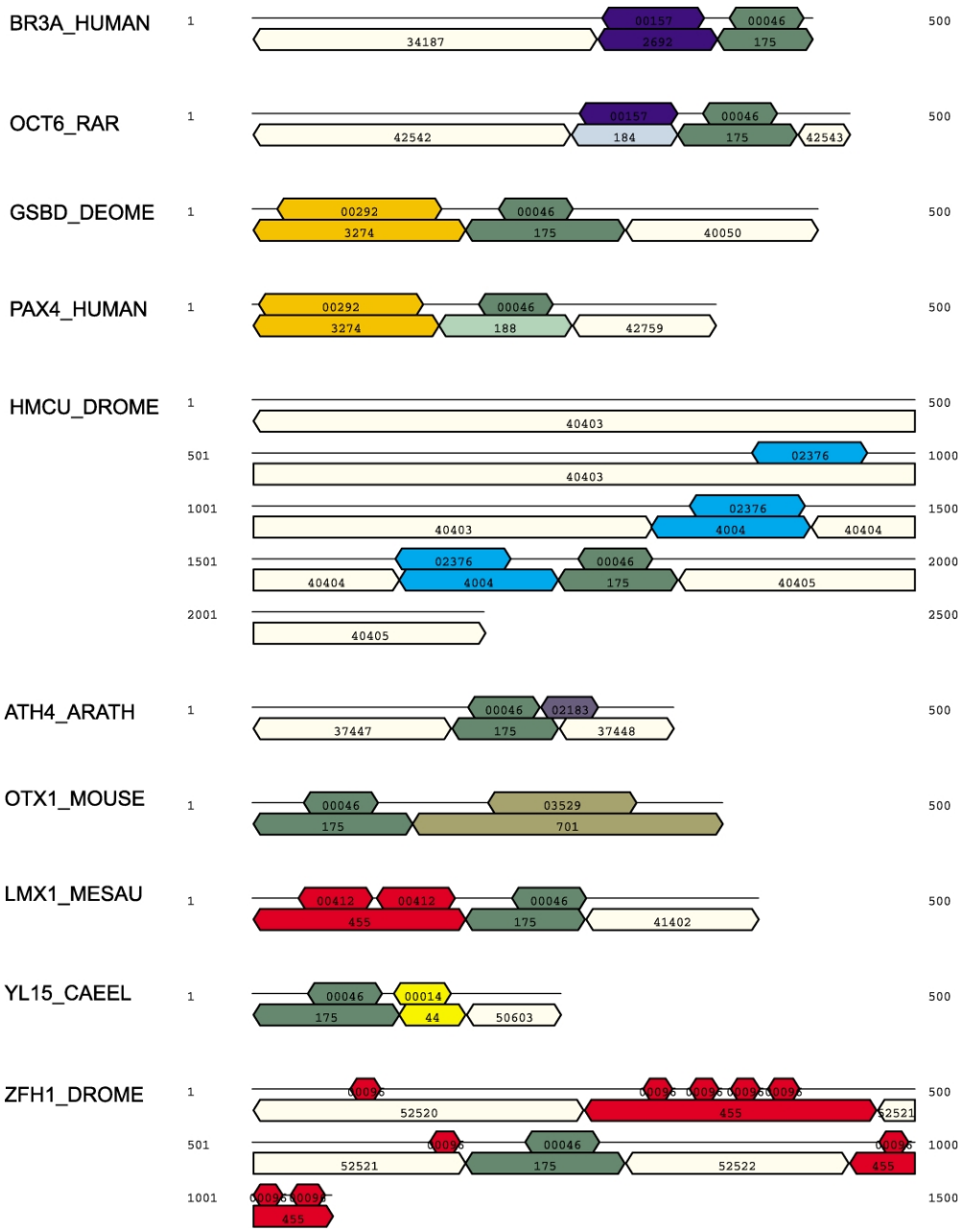

Figure 3. Homeobox domains in various proteins. Shown here are multi-domain proteins that contain a homeobox domain and other types of domains. The PFAM domain definitions are shown at the top of each sequence, ADDA domain definitions are below. Note the complete coverage of the proteins by ADDA domains and the overlap with PFAM where the latter are defined. Colours: green/light-green, PF00046 (Homeobox); red, PF00412 (LIM), PF00096 (C2H2 zinc finger); blue, light-blue, PF00157 (POU); orange, PF00292 (PAX); navy, PF02183 (leucine zipper); olive, PF03529 (Otx1 transcription factor).

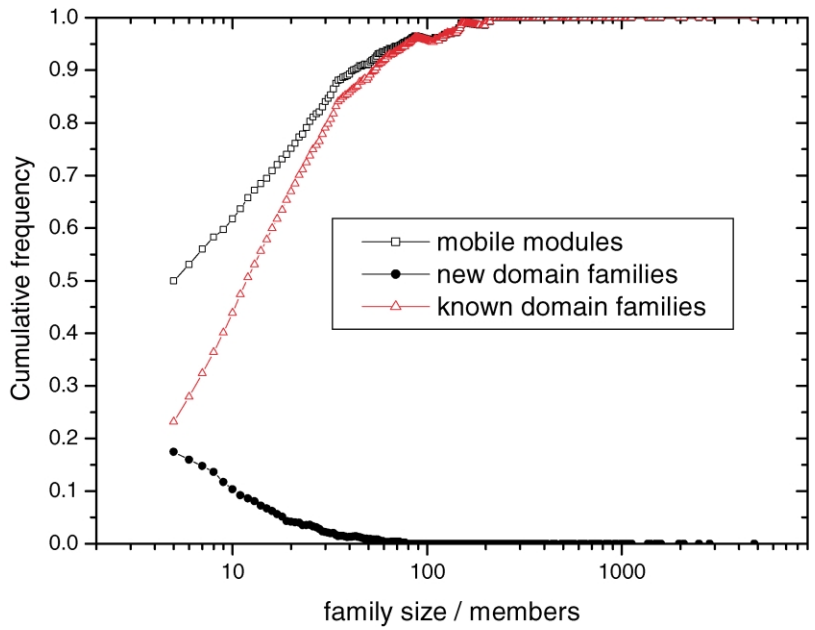

Figure 4. ADDA complements manual domain definitions. Domain families have been sorted by size. Shown here is a cumulative histogram of mobile domain families (squares) and domain families known to PFAM/SMART (circles). Coverage of large domain families by SMART and PFAM is complete, while ADDA defines many new families of smaller size.

PFAM, but with the benefit of defining domains for every residue in each sequence. Single instances of repeats are occasionally missed, for example, zinc-fingers in sequence ZFH1_DROME and a CUT domain in HMCU_DROME. These domains are detected using a repeat filtering algorithm[17] (data not shown).

Mobile modules

The global classification of ADDA allows us to systematically survey the set of all proteins and protein domains. Here, we concentrate on mobile modules, i.e. protein domain families that can appear in different sequence contexts. As a practical definition we adopt the following: a mobile module occurs in at least two multi-domain proteins with at least one domain each that is not shared with the other. All domain families are required to have at least five members in *nrdb40*.

Using this definition, we obtain 4230 families of mobile modules. The set of mobile modules encompasses virtually all domain families that have at least 100 members (Figure 4); only 13 domain families have more than 100 members and do not occur in conjunction with other domains. There are 115,273 sequences (48%) that contain at least one mobile module and 33,227 sequences (14%) that contain at least two mobile modules. Residue coverage by mobile modules is 47%.

Multi-domain proteins define associations between domain families. As has been observed previously,[18,19] the network of associations between domain families is dense and exhibits a scale-free degree distribution (data not shown). The largest component contains 21,062 domain families (62%). Removing all mobile modules decomposes the network leaving only 30 domain families in the largest component. We conclude that the set of mobile modules as defined above is complete.

Annotation of domain families

Domain families by ADDA have been annotated using PFAM[20] and SMART.[21] A domain family of ADDA is "known", if it contains at least five domains with annotations in PFAM and/or SMART: 3554 ADDA clusters are thus annotated and known to PFAM and/or SMART. A domain family is "new" if it contains no annotation from PFAM or SMART at all.

PFAM and SMART have concentrated on large domain families (Figure 4). Large ADDA families of size 50 or more (with only six exceptions) are all known to PFAM and SMART. Coverage of large domain families by these databases is thus complete. Domains annotated by PFAM/SMART tend to be taxonomically universal, i.e. they occur in all domains of life and many have structures associated with them (Table 2). This is consistent with the working principles of manual domain databases: large and universal families are likely to have drawn attention to them and many sequence domain families are defined around structural domains.

Table 2. Mobile modules defined by ADDA

Modules	Total	Structures	Universal	Domain specific	Species specific
All	420	791 (21%)	715 (17%)	1858 (44%)	266 (6%)
Known	1962	724 (37%)	627 (32%)	712 (36%)	166 (9%)
Novel	1476	37 (3%)	58 (4%)	1038 (70%)	70 (5%)

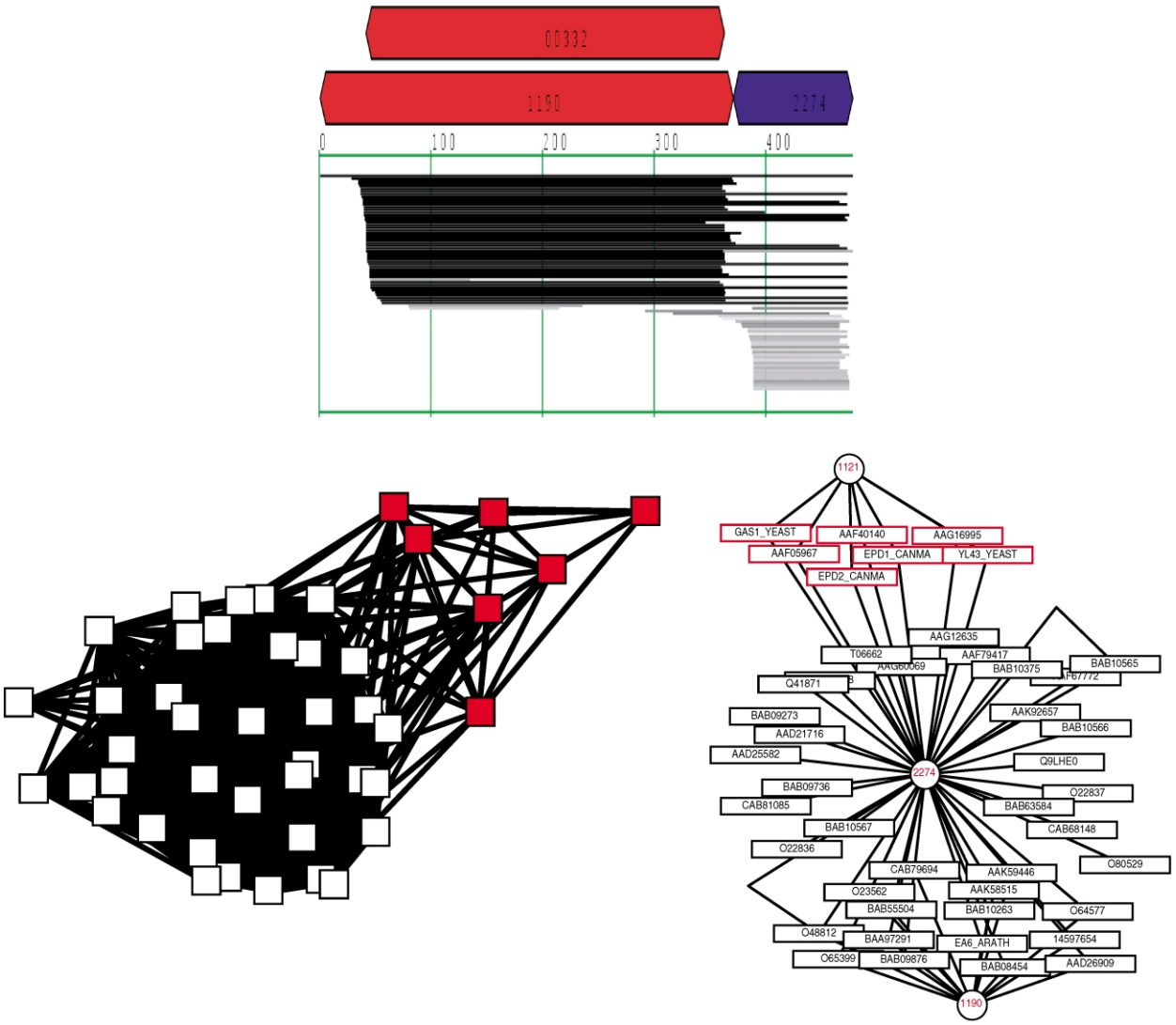

Figure 5. New domain family 2274. Members of the two subfamilies are differently coloured. Top: domain decomposition of sequence EA6_ARATH from *Arabidopsis thaliana* (top, PFAM; middle, ADDA; bottom, BLAST neighbours). Domain 2274 is blue. Bottom left: pair-wise BLAST alignments between members of domain family 2274. Bottom right: domain family 2274 is associated with domain families 1121 (PFAM: PF03198, glycolipid anchored surface protein (GAS1)) and domain family 1190 (PFAM: PF00332, glycosyl hydrolases family 17).

Novel domain families

ADDA extends PFAM and SMART by 1476 small to medium-sized mobile modules. The majority of these novel domain families have less than 50 members in *nrdb40*. In contrast to large domain families, the new domain families are mostly specific for a single domain of life (Table 2).

Structural coverage of new domain families is low (3%). An additional 25 domain families out of 37 new families with structures have been described by SCOP (68%). Missing from SCOP are families 2455 and 8080 that define sequence around single-domain structures 1K5O and 1MWR-A, respectively. The remaining families overlap with structures due to various artefacts, for example, spurious mapping of small peptides.

Typically, unknown mobile modules are located in sequences from genome projects and thus have no experimental information attached. Annotation has to be derived from other sources. Occasionally, domain families can be annotated through their association with known domains in multi-domain proteins.[22] Here, 1373 (93%) of the new mobile modules associate with a "known" domain family.

For example, family 2274 is a domain family specific to Eukaryotes that is associated with domains of known function (Figure 5). The 43 members of this family (*nrdb40*) fall into two subclasses that associate with two different types of glycoside hydrolases (families 17 and 72 according to the CAZy[23] database). The multiple alignment reveals six conserved cysteine residues (Figure 6).

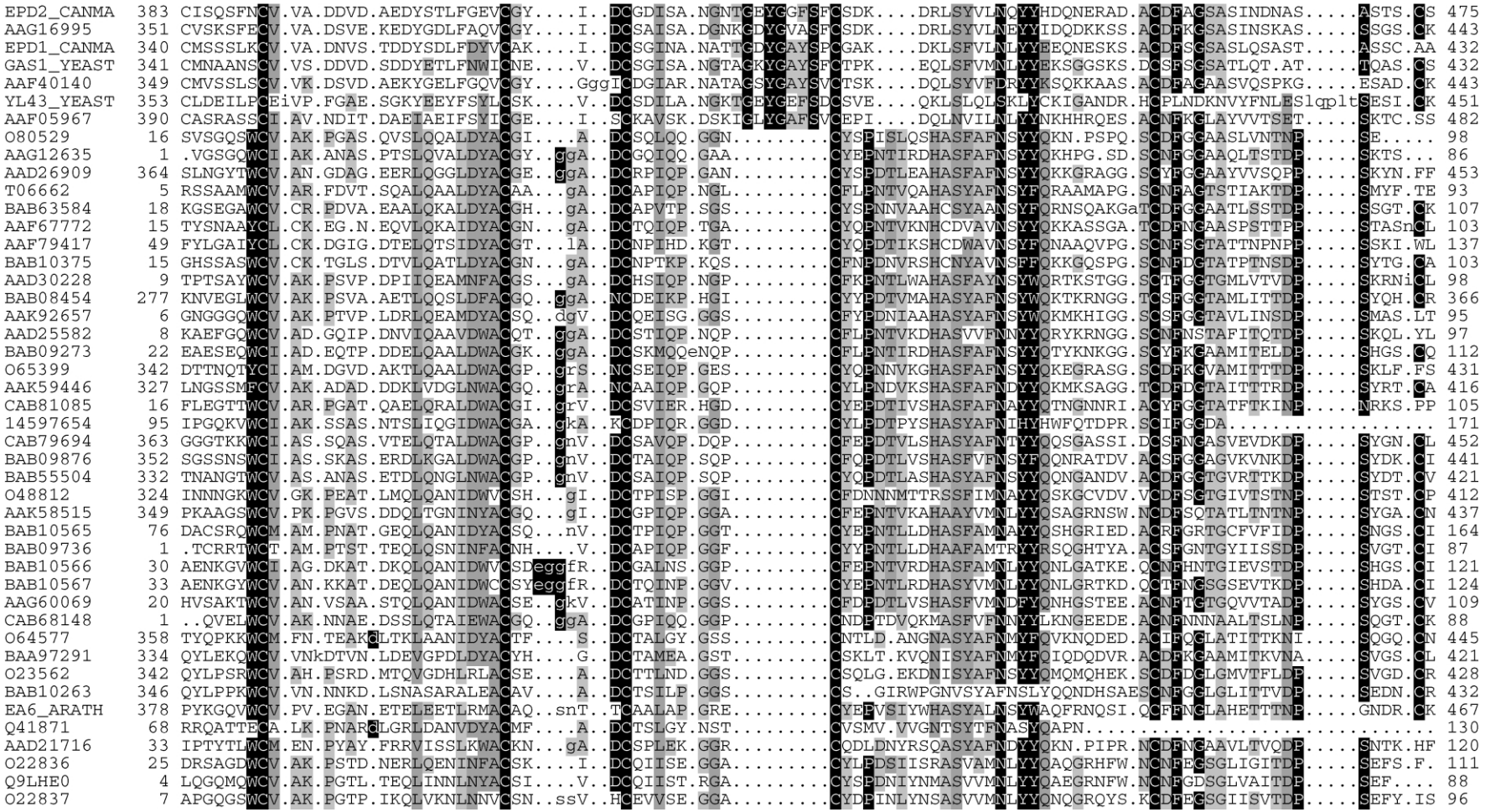

Figure 6. Multiple alignment of members in new domain family 2274.

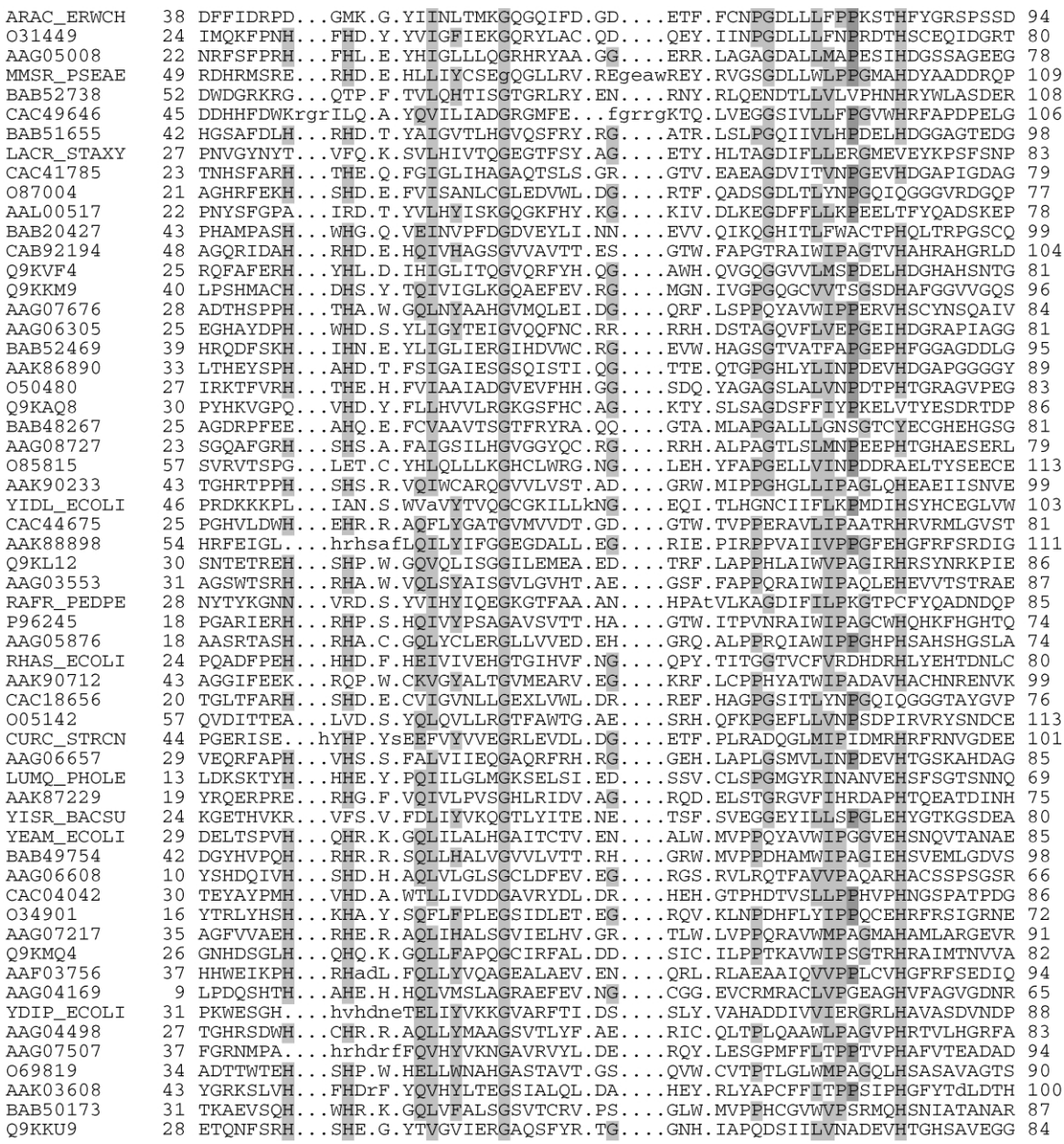

Figure 7. Multiple alignment of members in new domain family 967 reveals the conserved pattern HxH*G*PxxxH.

This domain might function as an extracellular binding domain.

In all, 792 modules contain less than five members with annotation from PFAM or SMART. The annotation might derive from unresolved domain splits by ADDA and/or wrong assignments by PFAM/SMART. Among these families are 321 small PFAM or SMART families, that ADDA extends substantially. Here, the ADDA family contains all members of the PFAM/SMART domain family and extends the latter at least twofold. For example, the arabinose operon regulating protein family (PFAM family PF02311) has five members in the Enterobacteriaceae. ADDA finds an additional 58 eubacterial sequences (cluster 967). A multiple alignment[24] reveals a conserved amino acid pattern HxH*G*PxxxH (Figure 7). Even though the three histidine residues and the proline are sequence distant, they form a tight cluster in the protein structure (Figure 8) suggestive of a metal binding site.

Validation

The domain families generated by ADDA are validated by comparison to manually curated domain definitions. For this purpose, we use SCOP,[25] which defines domains based on structures, and PFAM-A,[20] which is based on manually curated multiple sequence alignments.

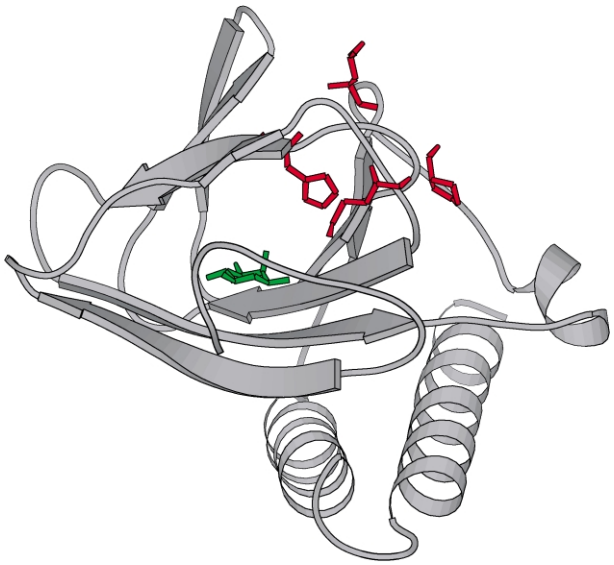

Figure 8. A putative metal binding site in domain family 976. The conserved pattern in Figure 7 (red residues, without the glycine) mapped onto the structure of the *Escherichia coli* gene regulatory protein AraC (Protein Data Bank identifier: 2AAC,[44] sequence identifier ARAC_ERWCH with D-fucose (green)). In this protein structure, the first two histidine residues have mutated to an asparagine and a methionine residue. The arrangement of the residues suggests a metal binding site in those members of the family where the histidine residues are present. The Figure was prepared using MOLSCRIPT.[45]

Quality of domain boundaries

In this section, we are interested in the accuracy of domain boundaries. The domain decomposition algorithm has two steps. In the first step, putative domain boundaries are created for each sequence in *nrdb40*. The putative domains of each sequence are organised hierarchically in a tree. Here, the benchmark is used to check whether the correct domain is contained in the set of putative domains. In the second step, domains are selected from the set of putative domain partitions. Here, the benchmark is used to check, whether the correct domains are selected based on our numerical criteria.

In the majority of all sequences the reference domain is among the putative domains (Figure 9). ADDA domains cover at least 90% of the residues in the reference domains for 87% of all SCOP domains and for 89% of all PFAM domains. Furthermore, most domains are of similar size, as the relative size between domains peaks distinctly at 100%.

An erroneous split occurs if there is no signal present in the multiple alignment that would allow us to define correct domain boundaries (by visual inspection). Three cases can be distinguished. Firstly, domain boundaries are defined based on limited data if a sequence has only few neighbours and thus the probability of error is high. Secondly, data in multiple alignments can be inconclusive if there is a continuum of possible domain boundaries. Finally, in some cases the alignment ends are clearly misleading. The latter frequently occurs with membrane proteins, as transmembrane regions have been masked before running BLASTP and thus alignments tend to terminate at transmembrane regions. This is an artefact of the generation of the sequence space graph and can be corrected in the future. Overall, these problem cases are rare. Merely 6% of PFAM and SCOP domains are covered by 70% or less by a putative domain, indicating that the reference domain is absent from the set of the putative domains.

In the second step of domain cutting, ADDA selects domains from the set of putative domains. Here, we verify that the correct domain or one that is larger is chosen from the set of putative domains.

In 92% (SCOP) and 93% (PFAM) of all cases ADDA selects domains from the set of putative domains that are of the same size or larger than the optimal domain. The distribution of relative sizes between selected domains and reference domains shifts towards larger domains (Figure 9). Sparse data are responsible for those cases where ADDA selects domains that are too short.

In conclusion, domain boundaries from ADDA correspond well to those in the reference domain definitions. In the cases where there is a discrepancy, ADDA errs mostly on the safe side and thus avoids over-fragmentation.

Quality of family definitions

Here, we are interested in the correspondence between clusters in ADDA as compared to the reference domain family classifications SCOP and PFAM. To this end, family labels of the respective reference classification are retrieved for all domains in an ADDA cluster. The cluster is then associated with the reference family to which most of its members match, all other matches are counted as contaminations.

Many ADDA clusters with reference domain annotation show "perfect" unification (100% sensitivity) with no contamination (100% selectivity) (Figure 10). These are 46% of all clusters when compared to SCOP and 41% when compared to PFAM. In all, 47% of all clusters in the case of SCOP and 61% of all clusters in the case of PFAM can be classified as "good" when we use a more permissive threshold, i.e. 90% sensitivity and selectivity.

Typically, large PFAM families are retrieved completely or almost completely (Table 3). For example, ADDA unifies 2288 out of 2307 PFAM protein kinase domains into a single cluster (99%, cluster 1). Similarly, ADDA assigns 5493 out of 5870 zinc fingers to the same cluster (94%, cluster 455), and classifies all 1640 ABC-transporters correctly (100%, cluster 22).

Occasionally, PFAM and SCOP super-family classifications disagree with ADDA. In many

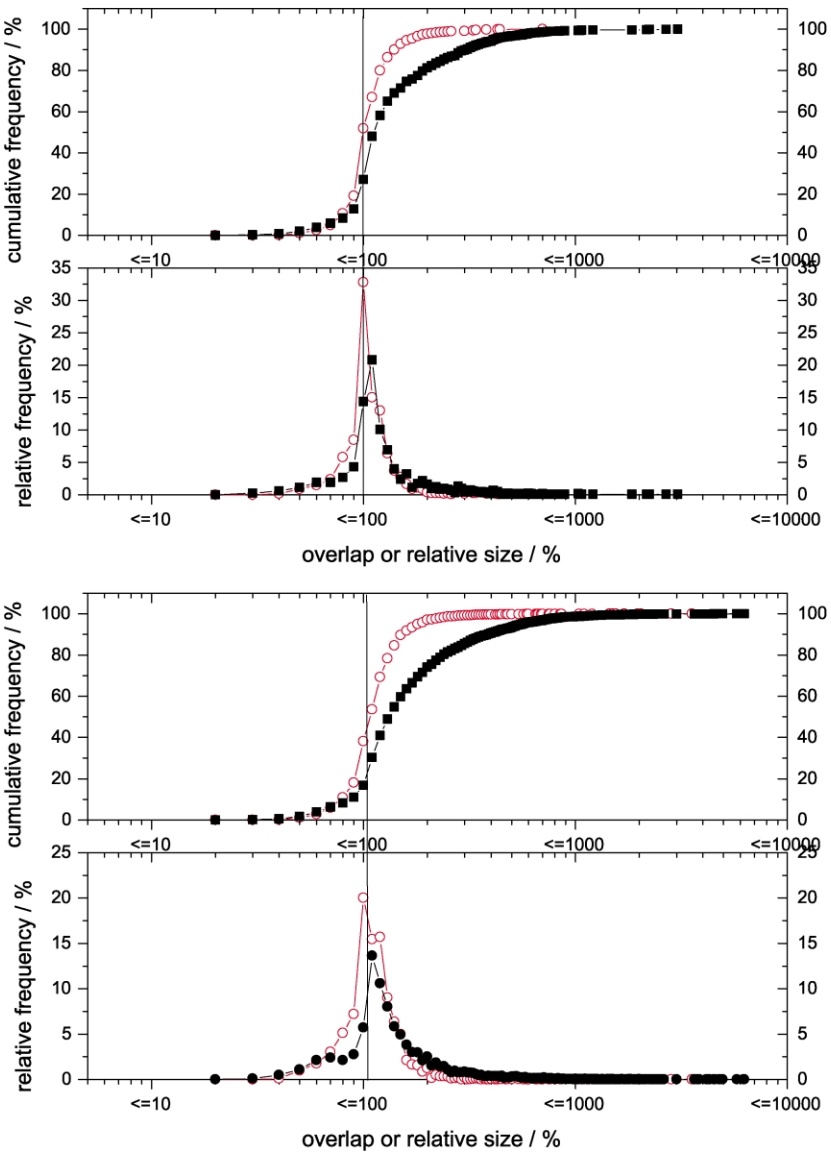

Figure 9. Comparison of ADDA domains to SCOP and PFAM domains. Putative domains were compared to SCOP (top) and PFAM (bottom) domain assignments and the overlap for incompletely covered reference domains and the relative size for completely covered reference domains (overlap = 100%) were recorded. Shown here are the histograms and cumulative histograms in bins of size 10% for best matching domains in the sets of putative domains (open circles) and those selected in the optimisation step (filled squares).

instances ADDA defines larger clusters than PFAM. For example, cluster 257 contains 1040 domains in 1016 sequences that are described as methyltransferases. While PFAM defines several sub-families, ADDA assigns all of them to the same class. The unification by ADDA is validated by structural similarities between methyltransferases.[26]

Cysteine-rich domains pose a special case. In ADDA, they are assigned to a single large cluster (cluster 44) encompassing EGF-domains (PF00008), Sushi domains (PF00084), several cysteine-rich repeats (PF03128 and PF02363), and others. Here, unification is based on sequence similarity, but falsely inferred homology: the cysteine residues result in a strong alignment signal even though these domains are not evolutionarily related. Rule-based post-processing might resolve this cluster into individual families.

Among the clusters that are neither "perfect" nor "good", most are concentrated along the axes of high selectivity but low sensitivity. In comparison to SCOP, 43% of all clusters fail to unify more than 90% of all members into a single SCOP family. Here, low sensitivity is a result of sequence diversity in SCOP super-families. Sensitivity compared to sequence-based PFAM classification is better: only 27% of all clusters unify less than 90% of all members of a PFAM family. In this group, low sensitivity is mostly due to regions masked due to predicted transmembrane segments or composition bias. The masking leads to truncated alignments that fall below the length and significance thresholds used in this work.

Clusters are contaminated by unrelated domains on a low level. Using SCOP as reference, 88% of all clusters contain only domains of a single SCOP super-family; with PFAM as reference, 75% of all clusters are completely pure. The latter lower

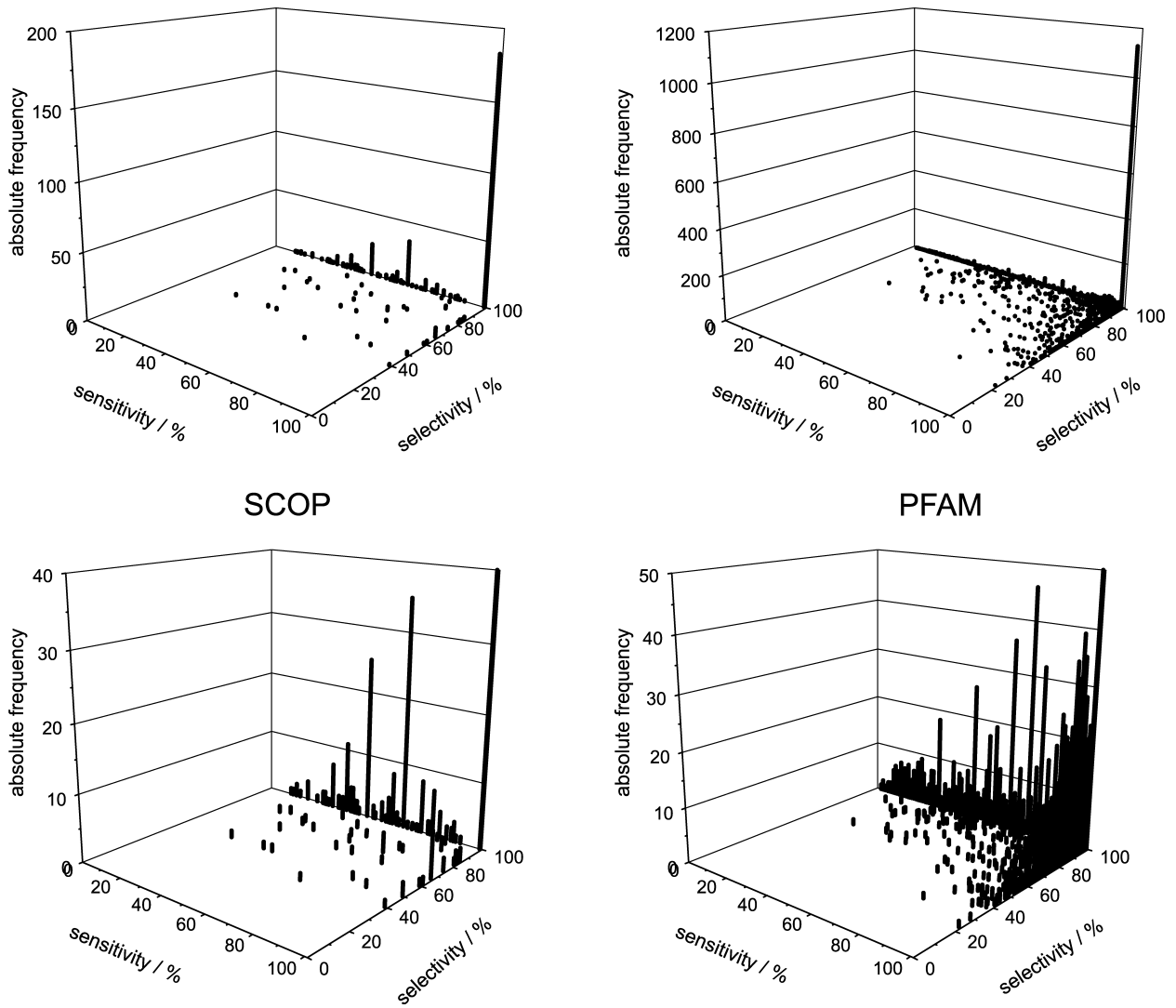

Figure 10. Histograms of sensitivity and selectivity of ADDA clusters compared to SCOP and PFAM. Each cluster is assigned a selectivity (cluster purity) and sensitivity (unification of associated reference domain family). The graphs in the bottom row are enlarged versions of those at the top (note vertical scale).

selectivity is mostly an artefact, as PFAM's family definitions are stricter than SCOP's and ADDA's super-family definitions. For example, methyltransferases are unified by ADDA into a single family (cluster 257, Table 3).

Contamination arises because domains are not resolved completely. The optimisation step selects domains conservatively, and thus some domains are not separated. This results in unrelated domains being unified into the same cluster. For example, the protein kinase cluster contains 2310 PFAM annotated domains, of which PFAM describes 2288 as kinases. Contamination is due to four G protein signalling domains, three domains each of types SH3, PH, and FHA, seven viral domains of various types, and eight domains in single copy numbers. The contamination in this cluster is less than 1%, which is typical for other large clusters as well.

Discussion

Here, we have presented a method for the complete decomposition of the sequence space graph into smaller components of domain families. While the original sequence space graph is dominated by one giant component, ADDA splits the sequences of *nrdb40* into 450,462 domains and assigns them to 33,879 non-singleton domain families.

The global classification by ADDA defines 1476 novel domain families which are evolutionarily mobile modules. The novel families tend to be small and specific to a single domain of life. The families thus complement those described in SMART and PFAM, which tend to focus on large and/or taxonomically universal families. Coverage of large domain families by these databases is complete.

Table 3. Selection of clusters in ADDA with PFAM annotation

Id	d_t^c	s_t^c	d_a^r	d_t^r	s_a^r	s_t^r	PFAM	Description
1	2830	2587	2288	2307	2242	2252	00069	Protein kinase domain
455	2497	1818	5493	5870	1004	1150	00096	Zinc finger, C2H2 type
22	2084	1703	1640	1640	1338	1338	00005	ABC transporter
67	1971	1371	1039	1122	1014	1084	00528	Binding-protein-dependent transport systems inner membrane component
30	1588	1416	630	636	614	617	00106	Short chain dehydrogenase
			203	551	119	266	00550	Phosphopantetheine attachment site
			83	83	83	83	01370	NAD-dependent epimerase/dehydratase family
51	1519	1436	709	728	700	717	00097	Zinc finger, C3HC4 type (RING finger)
			102	153	76	109	00643	B-box zinc finger
			87	87	52	52	01485	IBR domain
9	1492	951	1288	1353	806	839	00076	RNA recognition motif. (also RRM, RBD, or RNP domain)
330	1404	1106	5736	5808	751	777	00560	Leucine-rich repeat
728	1377	1072	4255	4588	886	918	00400	WD domain, G-beta repeat
629	1366	727	582	698	575	691	00702	Haloacid dehalogenase-like hydrolase
			268	279	266	274	00122	E1–E2 ATPase
962	1123	697	1664	2362	497	838	00047	Immunoglobulin domain
352	1112	915	691	855	690	849	00271	Helicase conserved C-terminal domain
			454	502	442	486	00270	DEAD/DEAH box helicase
			214	223	193	200	00176	SNF2 family N-terminal domain
18	1102	898	1087	1247	392	466	00036	EF hand
			362	369	247	252	00168	C2 domain
668	1068	1042	896	1039	892	1035	02518	Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase
			578	592	577	591	00512	His kinase A (phosphoacceptor) domain
			156	480	156	441	00672	HAMP domain
410	1050	907	648	656	629	635	00072	Response regulator receiver domain
			197	203	197	203	00196	Bacterial regulatory proteins, luxR family
257	1040	1016	92	92	92	92	01209	ubiE/COQ5 methyltransferase family
			47	47	47	47	00398	Ribosomal RNA adenine dimethylase
			43	59	43	59	01728	FtsJ-like methyltransferase
			30	34	30	34	02475	Met-10 + like-protein
			27	27	27	27	03141	Putative methyltransferase
			24	24	24	24	01135	Protein-L-isoaspartate(D-aspartate) *O*-methyltransferase (PCMT)
			16	16	16	16	01269	Fibrillarin
			15	16	15	16	02353	Cyclopropane-fatty-acyl-phospholipid synthase
220	993	675	386	465	378	455	00665	Integrase core domain
			54	84	54	74	00385	Chromo′ (CHRromatin Organization MOdifier) domain
381	942	752	565	585	469	482	00004	ATPase family associated with various cellular activities (AAA)
			87	87	87	87	00158	Sigma-54 interaction domain
			73	81	73	81	02954	Bacterial regulatory protein, Fis family
732	934	797	2130	2552	516	629	00515	TPR domain
107	919	896	244	246	244	246	00392	Bacterial regulatory proteins, gntR family
			194	203	194	203	01047	MarR family
			126	141	125	139	01022	Bacterial regulatory protein, arsR family
			40	94	40	94	03099	Biotin/lipoate A/B protein ligase family
			28	29	28	29	02237	Biotin protein ligase C-terminal domain
444	867	774	379	418	377	416	00561	Alpha/beta hydrolase fold
197	854	702	3394	3518	622	647	00023	Ankyrin repeat
239	831	586	464	948	461	935	00361	NADH-ubiquinone/plastoquinone (complex I), various chains
			149	256	149	256	00662	NADH-ubiquinone oxidoreductase (complex I), chain 5 N terminus
			74	74	74	74	01010	NADH-Ubiquinone oxidoreductase (complex I), chain 5 C terminus
175	830	806	748	909	729	871	00046	Homeobox domain
468	827	762	543	554	535	546	00535	Glycosyl transferase
			231	255	73	82	00652	QXW lectin repeat

The full Table is available at ftp://ftp.ebi.ac.uk/pub/contrib/heger/adda. Id, ADDA cluster number; d/s, number of domains/sequences; a/t, annotated domains/total number of domains in *nrdb40*; c/r, cluster/reference (PFAM). Low-level contamination (annotations with less than 10% of members than the major PFAM family) has been omitted.

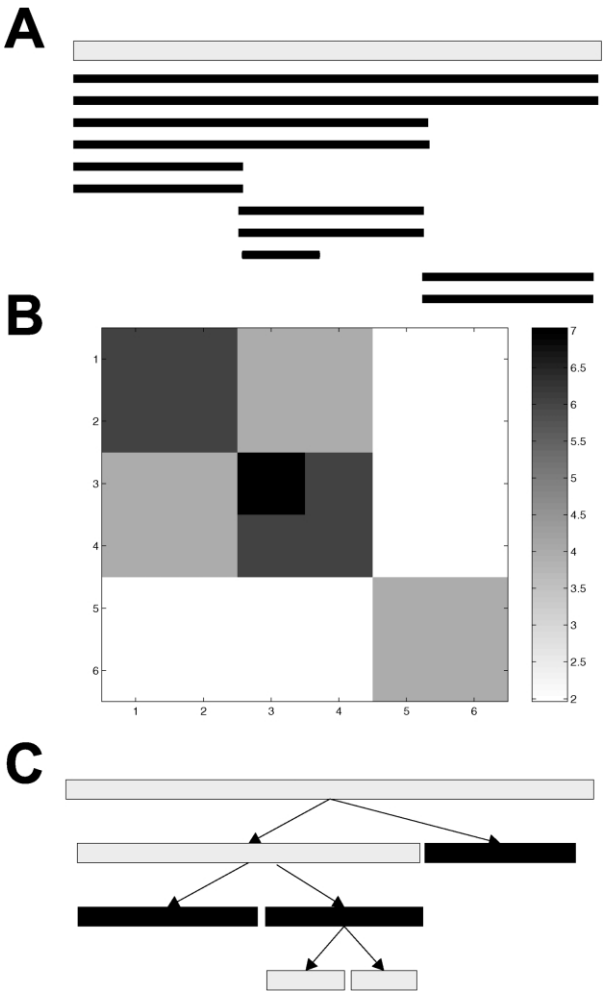

Figure 11. Hierarchical cutting of sequences. A, A typical query sequence of three domains with local alignments to its neighbours. The three domain boundaries are clearly visible; the middle domain contains a conserved motif. B, The residue column correlation matrix. C, The domain tree of putative partitions. The optimal partition (black) involves domains from different levels in the tree.

The concepts behind ADDA are simple and provide a clear definition of a domain, since the decomposition is based on an explicitly formulated model. The objective function is essentially parameter free, as parameters are determined based on a reference domain definition. Noise due to fragments and motif alignments is directly incorporated into the model. The method could thus be applied directly as part of a "high-throughput pipeline" to the output of genome sequencing projects.

ADDA is fast. Many of the steps can be run in parallel. The decomposition of the *nrdb40* graph takes about 48 hours wall-clock time using six standard PCs. The clustering can thus keep pace with the stream of sequences coming in from the genome sequencing projects. The domain families will be regularly updated and a web-server is currently under development.

Domains and families correspond well to reference domain definitions from PFAM and SCOP. Sequences are cut conservatively with a bias towards large domains. Here, ADDA errs on the safe side. A complete decomposition of the sequence space graph into families is achieved with a minimum number of domains, avoiding over-fragmentation of sequences.

Family unification rests upon a chain of reasoning involving the transitivity of homology. This is reasonably safe, since domain chaining has been eliminated by cutting sequences into domains beforehand. Homology is inferred by pairwise comparison between local neighbourhoods of putatively related domains that are close in sequence space. In this way clusters are built based on a chain of reasoning involving small steps of high confidence. We have demonstrated that even diverse families can be unified accommodating both spherical and elongated clusters in sequence space. This approach abandons the idea of generating a generalised profile model for a family. With profile models, families of different diversity require family-specific thresholds to achieve maximal discriminatory power.[20]

Limitations

Currently, ADDA is restricted to domains with sizes of more than 30 residues to avoid unification based on motifs, for example, P-loop-containing nucleotide-binding domains. Furthermore, the current sequence positional resolution of domain boundaries is ten residues. Thus, domain boundaries might be inaccurate for small domains or they might be missed completely.

Repeats are not resolved into individual domains but are annotated as repeat-containing regions. We are working on a version where these problems are solved by removing repeats beforehand.[17] Furthermore, ADDA assumes that a domain is continuous and thus does not accommodate inserted domains or circular permutations of domains. These oddities are usually subsumed under a single domain.

If there is no convincing signal in the BLASTP multiple alignments, domain boundaries are not resolved by ADDA. This leads to a low-level contamination of large clusters with unrelated domains. Post-processing might resolve the remaining conflicts. In particular, we plan to use context information of aligned sequence segments to resolve the few remaining domain conflicts.[27]

Methods

Sequence space graph

Sequence sets

Protein sequences from several sequence databases (SWISS-PROT and TREMBL,[28] PIR,[29] PDB,[30] WORMPEP, ENSEMBL[31]) were compiled into a single database. Data-

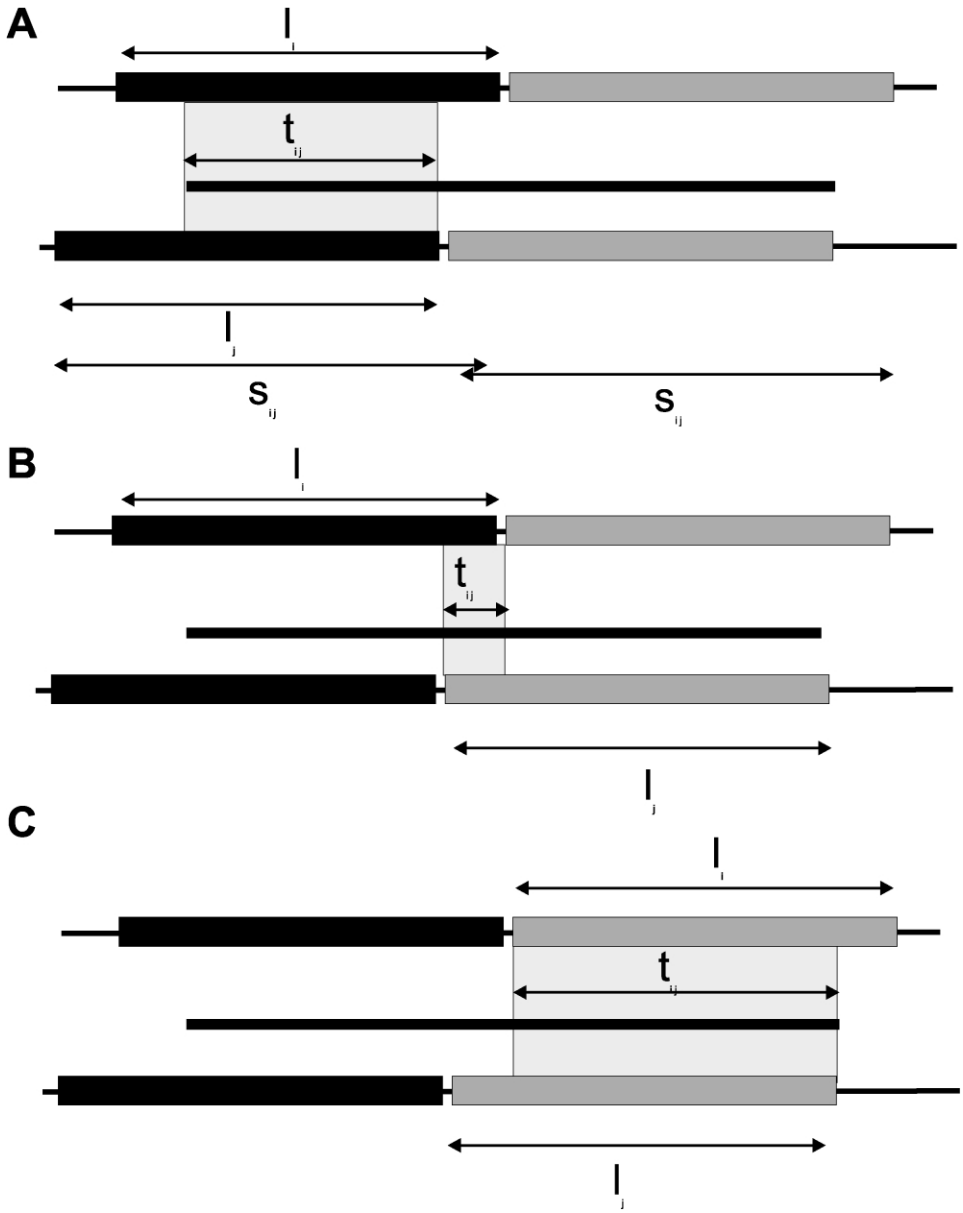

Figure 12. Comparison between two sequences (thin lines) sharing an alignment (thick line). The two sequences (lines) are partitioned into two domains each and the alignment is split into three segments. Thus, the likelihood function $L_{u,a,b}$ for this pair of sequences has three terms.

bases were current as of October 2001. Redundant sequences were removed using the programs nrdb[32] (sequence set *nrdb*, 782,238 sequences) and subsequently nrdb90[33,34] (sequence set *nrdb90*, 420,648 sequences). Sequences were masked for composition bias (Casari *et al.*, unpublished, similar to Promponas *et al.*[35]), transmembrane regions,[36] coiled-coils regions,[37] and short ungapped repeats.[17]

Pairwise alignments

All-on-all alignments for *nrdb90* were obtained by BLASTP.[38] Sequence masking by BLASTP was turned off. All hits with an *E*-value of less than 1.0 were kept. The results list was limited to 5000 matches and the reference size of the database for calibrating *E*-values was set to 6.5×10^7, otherwise default parameters were used: 240 Mio alignments were collected.

The result of all-*versus*-all alignments can be represented in a graph with sequences at its vertices and alignments at its edges. The *nrdb90* graph contained one major connected component of 367,482 sequences (87%) and 40,818 singletons (10%). Redundancy in the graph was reduced by removing sequences with more than 40% identity[16] (sequence set *nrdb40*). The resulting *nrdb40* graph contained 249,264 vertices, 25 Mio edges, a major component with 185,906 sequences (75%), and 50,986 singletons (20%). The *nrdb40* graph was used for clustering.

Eliminating edges of low confidence fragmented the sequence space graph but at the same time split domain families into disconnected components. For example, at

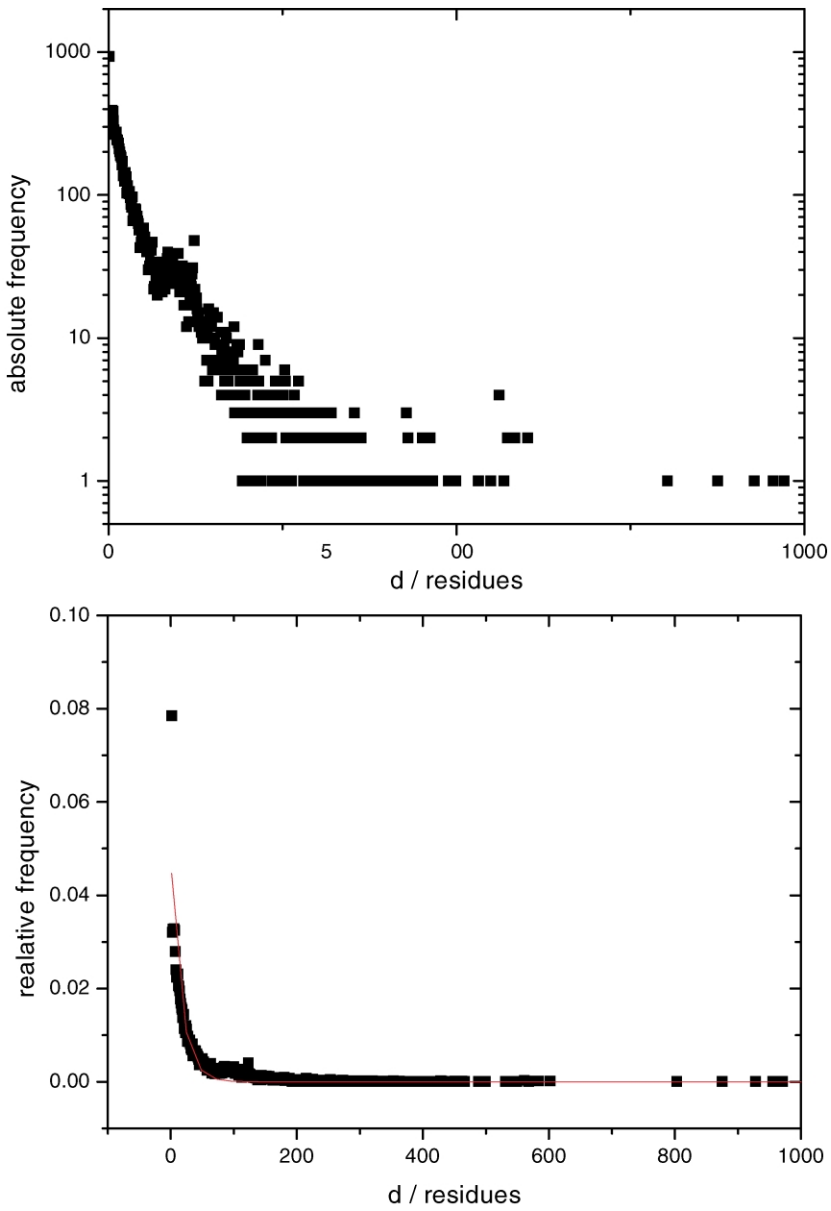

Figure 13. Probability of truncated alignments. Top, distribution of residues in SCOP domains that are not covered by alignments in the sequence space graph *nrdb40*. Bottom, fit of a first-order exponential decay function to relative frequencies. The equation of the fitted line is $P(d) = 0.05\,e^{-0.06d}$ ($\chi^2 = 4.95413^{-6}, R^2 = 0.89026$).

an *E*-value threshold of 1.0, 95% of all PFAM-families were contained in the same component, while 5% were distributed over more than one component. After removing the least significant edges (BLASTP *E*-value larger than 10^{-5}), already 22% of all PFAM-families were distributed over several components. Thus, generating the sequence space graph at an *E*-value threshold of 1.0 was a necessary requirement for unification.

Reference domain annotations

Domain definitions from SCOP 1.57[25] (super-family level), PFAM 7.3,[20] PRODOM 2001.3,[13] DOMO,[14] and SMART 3.4[21] were mapped onto sequences in *nrdb* and transferred onto sequences in *nrdb40* using BLASTP alignments.

Domain cutting

Domain cutting was a two-step procedure. In the first step, the algorithm generated a set of nested putative domain boundaries for each sequence in *nrdb40*. In the second step, it selected optimal domains for all sequences simultaneously.

Putative domain boundaries

Putative domain boundaries were derived for each sequence in a hierarchical manner yielding a tree of putative domains. Putative domain boundaries were determined based on the residue correlation matrix C. Entry (i, j) in the residue correlation matrix contained the number of protein neighbours that aligned to both columns i and j in the query sequence (Figure 11B). The residue correlation matrix was compressed by a factor of 10 for computational reasons, which limited the resolution of domain boundaries on the sequence to ten residues.

Based on the correlation matrix C a new domain boundary was defined. The new domain boundary split the sequence into two, and at the same time partitioned the symmetric correlation matrix into three sub-matrices. $C11$ and C^{22} measured the intra-domain correlation of

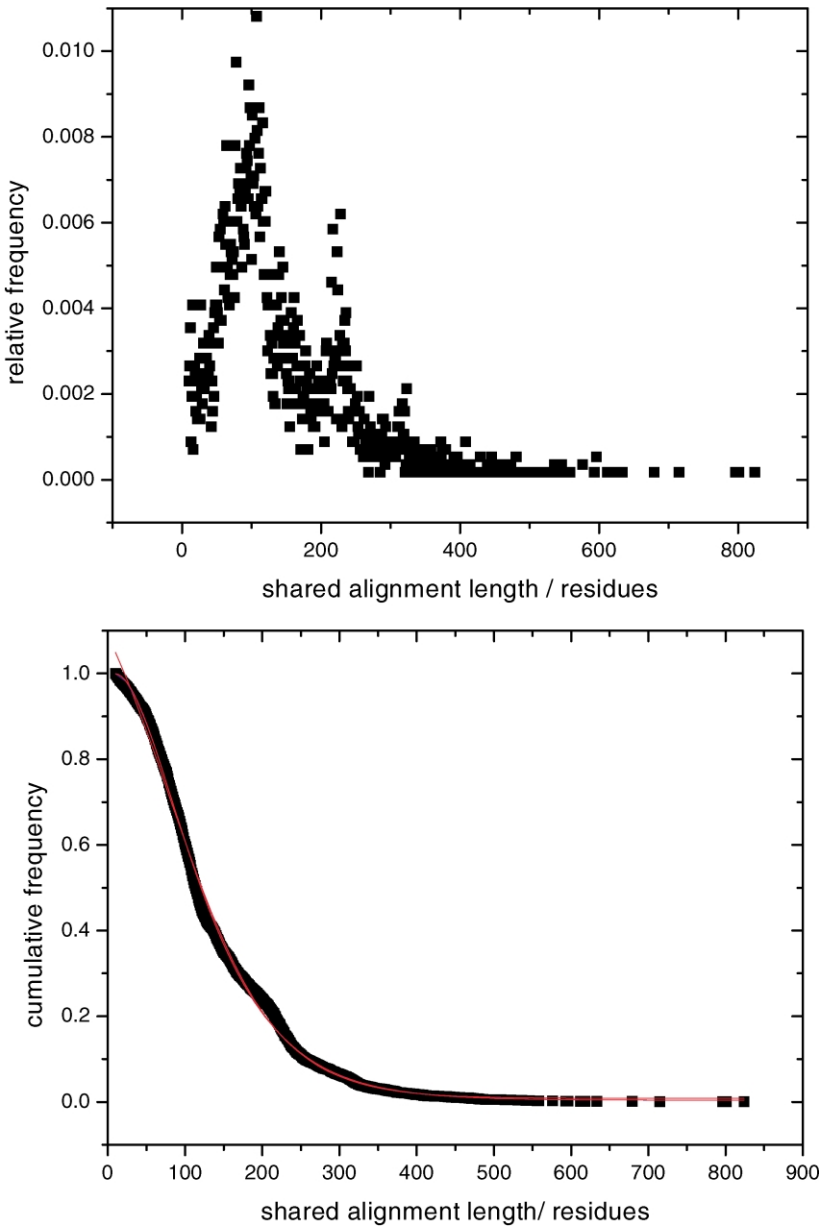

Figure 14. Probability of splitting an alignment. (Top) Distribution of the segment length common to a pair of SCOP domains and a linking alignment. (Bottom) Cumulative distribution and fit of an extreme value distribution. The equation of the fitted line is:

$$S(z) = 0.004 + 0.996\, e^{-e^{-z}-z+1},$$

$$z = \frac{t - 8.578}{76.411}\ (\chi^2 = 2.3^{-4},$$

$$R^2 = 0.998)$$

aligned neighbours while I quantified the inter-domain correlation between the two domains:

$$C = \begin{Vmatrix} C^{11} & \vdots & I \\ \cdots & & \cdots \\ I & \vdots & C^{22} \end{Vmatrix} \quad (1)$$

The domain boundary was placed so that intra-domain correlation was maximised and inter-domain correlation minimised using the χ^2 statistic for a two-by-two contingency table:

$$\chi^2 = \frac{(c^{11\times}c^{22} - i \times i)^2}{(c^{11} + i)^2(c^{22} + i)^2} \quad (2)$$

$$c^{11} = \sum_{i,j} C^{11}_{i,j} \quad (3)$$

$$c^{22} = \sum_{i,j} C^{22}_{i,j} \quad (4)$$

$$i = \sum_{i,j} I_{i,j} \quad (5)$$

in other words, splits were positioned at those residues in the sequence where the confidence was higher that two distinct domains were present.

Once a domain boundary was defined, further domain boundaries were defined by splitting sub-matrices C^{11} and C^{22}, and so on, until χ^2 was zero, or both resulting domains were less than 30 residues long. The result was a set of nested putative domains organised in a tree (Figure 11C).

Optimisation

The second stage of the algorithm selected domains from the sets of putative domains that were generated

for each sequence in the previous step. The selection was based on an objective function that modelled the block structure of BLASTP multiple alignments (see Introduction). The objective function was a likelihood-function that determined the likelihood of observing a specific pair of domains in two sequences sharing an alignment.

The likelihood $L_{u,a,b}$ between a pair of sequences a and b and a given partition u was given by:

$$L_{u,a,b} = \prod_{i,j} Q(t_{ij})R(d_i)R(d_j) \quad (6)$$

Indices i and j iterated over all possible combinations of domains mapping to an alignment between the two of sequences (Figure 12). t_{ij} was the length of the common segment between two domains and the alignment and $d_i = l_i - t_{ij}$ and $d_j = l_j - t_{ij}$ were the number of residues in domains i and j, respectively, not covered by the alignment. $Q(t)$ gave the probability of splitting an alignment of length smaller than or equal to t and $R(d)$ the probability of leaving d residues not covered. The distributions of $Q(t)$ and $R(d)$ were determined from data (see following sections).

The total likelihood for all sequences and a partition u was given by the product of $L_{u,a,b}$ over all pairs of sequences that were linked by an alignment.

Probability of truncated alignments

Truncated alignments leave residues in a domain unaccounted for by the alignment. The probability $R(d)$ that d residues in a domain are not covered by an alignment was modelled by an exponential decay density function:

$$R(d) = a\, e^{-ad} \quad (7)$$

with a being the single free parameter. The exponential decay function was used because of its memory-less property. It awarded the same penalty irrespective of the exact location of the alignment within a domain.

The free parameter a was determined by fitting data from a reference domain definition and a sequence space graph to a decay function. Here, the non-redundant set of sequences *nrdb40* containing SCOP domains was used. For every pair of sequences sharing a common domain and an alignment, d was determined. The resulting distribution was exponential (Figure 13).

Probability of split alignments

The block model of BLASTP multiple alignments assumed that alignments corresponded to full-length domains. However, multi-domain proteins cause alignments to be split at domain boundaries. $Q(t)$ modelled this as a probability of segmenting an alignment shorter than or equal to t:

$$Q = P(\{ \geq 1 \text{ cuts in alignment of length } T \leq t\}) \quad (8)$$

Q was a cumulative distribution where the event of not cutting an alignment had a higher probability than the event of cutting an alignment. The choice was motivated by the desire to have no *a priori* assumption over domain lengths. Q was estimated by using its complementary distribution S:

$$S = 1 - Q = P(\{\text{no cuts in alignment of length } T \leq t\}) \quad (9)$$

S was calculated from the same dataset as in the previous section; it was the cumulative frequency distribution of domain-fragmented alignments of length t. The probability S was modelled as an extreme value distribution with four free parameters (Figure 14):

$$S = p_0 + A\, e^{-e^{-z}-z+1} \text{ where } z = \frac{t - t_c}{w} \quad (10)$$

Validation of parameters

ADDA was robust with respect to parameters for both P and Q. In ten iterations, 10% of randomly selected SCOP super-families (and all associated domains) were eliminated from the set used for parameter fitting. ADDA was then run with different parameter sets on sequences containing domains not used for fitting that particular set of parameters. The results were compared amongst common sequences in the ten sets. In all cases, ADDA produced identical domain definitions.

Optimisation strategy

The space of all possible domain partitions of all sequences in *nrdb40* was too large to enumerate exhaustively. Therefore, a greedy optimisation strategy was used. Initially, all n sequences were uncut providing n domains, i.e. the optimisation procedure started at the top of the trees containing the putative domains. The algorithm then iterated over the list of all domains and split each in turn according to the pre-computed trees. This step corresponded to descending one level in the tree. If the likelihood of the new partition increased with respect to the previous partition, the split was accepted and the original domain was replaced with its two children. The algorithm repeatedly iterated over all domains until convergence was achieved, i.e. no additional cut in any domain increased the likelihood. This heuristic did not guarantee to find the exact location of the global optimum, but as sequences were initially uncut, the bias was towards long domains.

Unification

The sequence space graph was converted into a domain graph based on the domains calculated in the previous step. In the domain graph each vertex corresponded to a domain and each edge to an alignment between domains. Edges were removed, if the alignment covered one of the domains by less than 20% of its length. Furthermore, if a domain on sequence A was linked to several adjacent domains on sequence B, the one domain in B which overlapped most with the domain in A was recorded, and all other edges were removed.

An edge between a domain i on sequence A and a domain j on sequence B was weighted by the relative overlap w_{ij} between the alignment and the two domains (Figure 12):

$$w_{ij} = 1 - \frac{t_{ij}}{s_{ij}} \quad (11)$$

The domain graph was decomposed into connected components. For each component a minimum spanning tree was calculated using Kruskal's algorithm.[39] Spurious links were removed at this stage by checking each alignment *via* profile–profile alignment (see below). Only $n - 1$ alignments had to be performed per tree with n vertices. The removal of edges left a new set of minimum spanning trees. Finally, each domain in the

same minimum spanning tree was assigned to the same domain family.

Profile–profile alignments

Profiles[3] were built from *nrdb90* neighbourhoods, regularised using a nine-component Dirichlet-mixture[40] and rescaled by a factor of 0.3. Profiles were aligned using the local alignment algorithm[41] with affine gap penalties of -10 and -1 for gap opening and gap elongation, respectively. The score $s(i, j)$ for aligning two profile positions i with j was given by the weighted sum over all amino acid types a:

$$s(i,j) = \sum_{\mathrm{a}} [p_i(\mathrm{a})s_j(\mathrm{a}) + p_j(a)s_i(\mathrm{a})] \qquad (12)$$

where $p_x(\mathrm{a})$ and $s_x(\mathrm{a})$ were the regularised relative frequencies[40] and the profile scores in column x and amino acid a, respectively.

Sensitivity and selectivity of the profile–profile alignment method was benchmarked with a "SCOP-test".[42,43] Domains of less than 40% sequence identity were retrieved from the SCOP database and aligned all-against-all. An alignment was classified as true positive, if the SCOP super-family labels of the two aligned segments matched, otherwise it was declared to be a false positive. The benchmark set contained 3098 domain sequences encompassing 792 different super-families. There were 25,859 true positive pairs and 4,771,394 true negative pairs.

Based on the SCOP-test a threshold-score of 83 was defined. At this score, the rate of incorrectly classifying a pair as homologous was 5%, while 18% of true positive pairs were detected. Note that in the application of the algorithm, ADDA tests mainly close relatives in the minimum spanning tree. Alignments with a score of less than the threshold were removed from the minimum spanning trees. Alignments with a score of more than 415 were accepted without checking. All other alignments were subjected to the calculation of a Z-score (number of standard deviations above the mean, 50 shuffles, threshold 5.0).

Validation of domain boundaries

Domain boundaries were validated against reference domain definitions from SCOP and PFAM. Each reference domain was matched to all putative domains and the maximal overlapping domain defined as the best matching domain. For each best matching domain the coverage of the reference domain was recorded. Repeated domains and domains containing transmembrane regions were omitted, because they cause artefacts that could be and will be removed in the future. Only sequences originating from SWISS-PROT were considered in order to avoid artefacts due to automatic gene prediction methods.

Validation of unification

Unification properties of ADDA were measured as selectivity and sensitivity with respect to the reference domain family classifications PFAM and SCOP. To this end, matches between ADDA domains and reference domains were recorded if they overlapped by at least ten residues. Each ADDA cluster was then associated with the reference domain family to which most of its members matched, the other matches were classified as contaminations to that cluster.

Selectivity was defined as cluster purity, i.e. an ADDA cluster was designated to be perfectly pure if its members matched exclusively to the associated reference domain family. Selectivity s_i of cluster i was given by $s_i = n_{ia}/n_i$, where n_{ia} was the number of domains in cluster i matching to the associated reference domain family a and n_i was the total number of domains in cluster i matching to any reference domain family. Cluster contamination was $c_i = 1 - s_i$.

An ADDA cluster achieved perfect sensitivity if it contained all members of a single reference domain family. Sensitivity, or equivalently, unification u_i of cluster i was defined as $u_i = n_{ia}/n_{\mathrm{a}}$ with n_{a} being the total number of domains of reference domain a in *nrdb40*.

References

1. Wetlaufer, D. B. (1973). Nucleation, rapid folding, and globular intrachain regions in proteins. *Proc. Natl Acad. Sci. USA*, **70**, 697–701.
2. Doolittle, R. F. & Bork, P. (1993). Evolutionarily mobile modules in proteins. *Sci. Am.* **269**, 50–56.
3. Gribskov, M., McLachlan, A. D. & Eisenberg, D. (1987). Profile analysis: detection of distantly related proteins. *Proc. Natl Acad. Sci. USA*, **84**, 4355–4358.
4. Eddy, S. (1998). Profile hidden markov models. *Bioinformatics*, **14**, 755–763.
5. Brazma, A., Jonassen, I., Eidhammer, I. & Gilbert, D. (1998). Approaches to the automatic discovery of patterns in biosequences. *J. Comput. Biol.* **5**, 279–305.
6. Vitkup, D., Melamud, E., Moult, J. & Sander, C. (2001). Completeness in structural genomics. *Nature Struct. Biol.* **8**, 559–566.
7. Liu, J. & Rost, B. (2002). Target space for structural genomics revisited. *Bioinformatics*, **18**, 922–933.
8. Grundy, W. N. (1998). Homology detection *via* family pairwise search. *J. Comput. Biol.* **5**, 479–491.
9. Yona, G., Linial, N., Tishby, N. & Linial, M. (1998). A map of the protein space—an automatic hierarchical classification of all protein sequences. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **6**, 212–221.
10. Heger, A. & Holm, L. (2000). Towards a covering set of protein family profiles. *Prog. Biophys. Mol. Biol.* **73**, 321–337.
11. Krause, A. & Vingron, M. (1998). A set-theoretic approach to database searching and clustering. *Bioinformatics*, **14**, 430–438.
12. Enright, A. J., Van Dongen, S. & Ouzounis, C. A. (2002). An efficient algorithm for large-scale detection of protein families. *Nucl. Acids Res.* **30**, 1575–1584.
13. Corpet, F., Gouzy, J. & Kahn, D. (1999). Recent improvements of the prodom database of protein domain families. *Nucl. Acids Res.* **27**, 263–267.
14. Gracy, J. & Argos, P. (1998). Automated protein sequence database classification. ii. delineation of domain boundaries from sequence similarities. *Bioinformatics*, **14**, 174–187.
15. Gould, S. & Eldregde, N. (1977). Punctuated equilibria: the tempo and mode of evolution reconsidered. *Paleobiology*, **3**, 115–151.
16. Park, J., Holm, L., Heger, A. & Chothia, C. (2000). Rsdb: representative protein sequence databases have high information content. *Bioinformatics*, **16**, 458–464.

17. Heger, A. & Holm, L. (2000). Rapid automatic detection and alignment of repeats in protein sequences. *Proteins, Struct. Funct. Genet.* **41**, 224–237.
18. Wuchty, S. (2001). Scale-free behavior in protein domain networks. *Mol. Biol. Evol.* **18**, 1694–1702.
19. Koonin, E. V., Wolf, Y. I. & Karev, G. P. (2002). The structure of the protein universe and genome evolution. *Nature*, **420**, 218–223.
20. Bateman, A., Birney, E., Cerruti, L., Durbin, R., Etwiller, L., Eddy, S. R. *et al.* (2002). The pfam protein families database. *Nucl. Acids Res.* **30**, 276–280.
21. Letunic, I., Goodstadt, L., Dickens, N. J., Doerks, T., Schultz, J., Mott, R. *et al.* (2002). Recent improvements to the smart domain-based sequence annotation resource. *Nucl. Acids Res.* **30**, 242–244.
22. Doerks, T., Copley, R. R., Schultz, J., Ponting, C. P. & Bork, P. (2002). Systematic identification of novel protein domain families associated with nuclear functions. *Genome Res.* **12**, 47–56.
23. Coutinho, P. & Henrissat, B. (1999). Carbohydrate-active enzymes: an integrated database approach. In *Recent Advances in Carbohydrate Bioengineering* (Gilbert, H., Davies, G., Henrissat, B. & Svensson, B., eds), pp. 3–12, The Royal Society of Chemistry, Cambridge, UK.
24. Heger, A., Lappe, M. & Holm, L. (2003). Accurate detection of very sparse sequence motifs. *Proc. Annu. Int. Conf. Res. Comput. Mol. Biol.* In press.
25. Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (1995). Scop: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* **247**, 536–540.
26. Holm, L. & Sander, C. (1993). Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.* **233**, 123–138.
27. Heger, A. & Holm, L. (2001). Picasso: generating a covering set of protein family profiles. *Bioinformatics*, **17**, 272–279.
28. Bairoch, A. & Apweiler, R. (2000). The swiss-prot protein sequence database and its supplement trembl in 2000. *Nucl. Acids Res.* **28**, 45–48.
29. Wu, C. H., Huang, H., Arminski, L., Castro-Alvear, J., Chen, Y., Hu, Z. Z. *et al.* (2002). The protein information resource: an integrated public resource of functional annotation of proteins. *Nucl. Acids Res.* **30**, 35–37.
30. Westbrook, J., Feng, Z., Jain, S., Bhat, T. N., Thanki, N., Ravichandran, V. *et al.* (2002). The protein data bank: unifying the archive. *Nucl. Acids Res.* **30**, 245–248.
31. Hubbard, T., Barker, D., Birney, E., Cameron, G., Chen, Y., Clark, L. *et al.* (2002). The ensembl genome database project. *Nucl. Acids Res.* **30**, 38–41.
32. Gish, W., (1997). nrdb—quasi-non-redundant database generator.
33. Holm, L. & Sander, C. (1998). Removing near-neighbour redundancy from large protein sequence collections. *Bioinformatics*, **14**, 423–429.
34. Li, W., Jaroszewski, L. & Godzik, A. (2001). Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, **17**, 282–283.
35. Promponas, V. J., Enright, A. J., Tsoka, S., Kreil, D. P., Leroy, C., Hamodrakas, S. *et al.* (2000). Cast: an iterative algorithm for the complexity analysis of sequence tracts. Complexity analysis of sequence tracts. *Bioinformatics*, **16**, 915–922.
36. Sonnhammer, E. L., von Heijne, G. & Krogh, A. (1998). A hidden markov model for predicting transmembrane helices in protein sequences. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **6**, 175–182.
37. Lupas, A., Van Dyke, M. & Stock, J. (1991). Predicting coiled coils from protein sequences. *Science*, **252**, 1162–1164.
38. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucl. Acids Res.* **25**, 3389–3402.
39. Kruskal, J. (1956). On the shortest spanning subtree of a graph and the travelling salesman problem. *Proc. Am. Math. Soc.* **7**, 48–50.
40. Sjolander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Mian, I. S. & Haussler, D. (1996). Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput. Appl. Biosci.* **12**, 327–345.
41. Smith, T. & Waterman, M. (1981). Indentification of common molecular subsequences. *J. Mol. Biol.* **147**, 195–197.
42. Brenner, S. E., Chothia, C. & Hubbard, T. J. (1998). Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc. Natl Acad. Sci. USA*, **95**, 6073–6078.
43. Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T. & Chothia, C. (1998). Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J. Mol. Biol.* **284**, 1201–1210.
44. Soisson, S. M., MacDougall-Shackleton, B., Schleif, R. & Wolberger, C. (1997). The 1.6 Å crystal structure of the arac sugar-binding and dimerization domain complexed with D-fucose. *J. Mol. Biol.* **273**, 226–237.
45. Kraulis, P. (1991). Molscript: a program to produce both detailed and schematic plots of protein structures. *J. Appl. Crystallog.* **24**, 946–950.

Edited by G. von Heijne

(Received 18 November 2002; received in revised form 20 February 2003; accepted 24 February 2003)