BIOINFORMATICS

Vol. 18 no. 12 2002
Pages 1666–1672

PFDB: a generic protein family database integrating the CATH domain structure database with sequence based protein family resources

Adrian J. Shepherd [1, 2,], Nigel J. Martin [2], Roger G. Johnson [2], Paul Kellam [3] and Christine A. Orengo [1]*

[1]Department of Biochemistry and Molecular Biology, University College, London, Gower Street, London WC1E 6BT, [2]School of Computer Science and Information Systems, Birkbeck College, Malet Street, London WC1E 7HX, and [3]Wohl Virion Centre, Department of Immunology and Molecular Pathology, Windeyer Institute of Medical Sciences, 46 Cleveland Street, London W1P 6DB, UK

Received on December 21, 2001; revised on May 8, 2002; accepted on May 16, 2002

ABSTRACT

Motivation: The PFDB (Protein Family Database) is a new database designed to integrate protein family-related data with relevant functional and genomic data. It currently manages biological data for three projects—the CATH protein domain database (Orengo *et al.*, 1997; Pearl *et al.*, 2001), the VIDA virus domains database (Albà *et al.*, 2001) and the Gene3D database (Buchan *et al.*, 2001). The PFDB has been designed to accommodate protein families identified by a variety of sequence based or structure based protocols and provides a generic resource for biological research by enabling mapping between different protein families and diverse biochemical and genetic data, including complete genomes.

Results: A characteristic feature of the PFDB is that it has a number of meta-level entities (for example *aggregation, collection* and *inclusion*) represented as base tables in the final design. The explicit representation of relationships at the meta-level has a number of advantages, including flexibility—both in terms of the range of queries that can be formulated and the ability to integrate new biological entities within the existing design. A potential drawback with this approach—poor performance caused by the number of joins across meta-level tables—is avoided by implementing the PFDB with materialized views using the mature relational database technology of Oracle 8*i*. The resultant database is both fast and flexible.

This paper presents the principles on which the database has been designed and implemented, and describes the current status of the database and query facilities supported.

Availability: http://bsmsn01.biochem.ucl.ac.uk/

*To whom correspondence should be addressed.

INTRODUCTION

The Protein Family Database (PFDB) is a new database that currently manages biological data for three projects—the CATH protein domain database (Orengo *et al.*, 1997; Pearl *et al.*, 2000), the VIDA virus domains database (Albà *et al.*, 2001) and the Gene3D database (Buchan *et al.*, 2001). Future additions to the database will include protein families identified for specific biological systems or organisms. The database is currently being extended to include protein families identified in the eye (EyeSite database, Slingsby *et al.*, personal communication). The PFDB provides a mechanism for integrating protein families identified using independent classification protocols.

Although structural data is more sparse with approximately only 13 000 protein structures currently determined compared to over 12 million sequences, structure is much more highly conserved within a protein family allowing more distant homologues to be more readily identified. The motivation for the PFDB is to integrate protein families identified using structure-based protocols with those determined using various sequence-based protocols together with all the available functional and genomic data for these families. Such integration enables queries between families and thereby facilitates mapping of individual structural domains onto sequence based families. Mapping is achieved using a common identifier (GenBank, NCBI, Benson *et al.*, 2000) where possible and using simple pairwise sequence alignment methods (e.g. FASTA) where necessary.

A key data resource underpinning the PFDB is the CATH domain structure database which currently contains some 1200 protein superfamilies identified using both sequence and structure based protocols (Pearl *et al.*,

2001). Individual domains are identified automatically using a consensus approach (Jones *et al.*, 1998) and a recently developed method that detects recurrent domains (Harrison *et al.*, 2002). Any ambiguous assignments are validated manually (Pearl *et al.*, 2001).

Originally a flat-file database of protein structural domains, CATH has expanded dramatically over the past 18 months to include large quantities of new data—notably sequence families derived from ~300 000 GenBank sequences, and genomic data from GenBank (Benson *et al.*, 2000). These have been identified using profile based methods and hidden Markov models (PSI-BLAST, Altschul *et al.*, 1997; IMPALA, Schäffer *et al.*, 1999; SAMt, Karplus *et al.*, 1998), and a DomainFinder algorithm (Pearl *et al.*, 2001) which determines the sequence region corresponding to a given structural domain.

Since CATH is so widely used within the biochemistry community, it is essential that the new data in CATH is managed effectively. Furthermore, there is increasing need to map between the structural families identified in CATH and other local protein family resources (e.g. VIDA, EyeSite) and future databases established within the MRC Cooperative which aims to use structural data to improve understanding of the molecular basis of disease. The decision was therefore taken to establish a generic protein family database, incorporating CATH, VIDA, EyeSite and other related database resources, using a database management system (DBMS).

The VIDA database contains a complete collection of homologous protein families derived from open reading frames from complete and partial virus genomes of particular virus families (currently herpesviruses, coronaviruses and arteriviruses). These are mostly sequence-based families that have been identified using a protocol based on the profile-based method (PSI-BLAST, Altschul *et al.*, 1997; MKDOM, Corpet, 1988). This method attempts to identify domains within gene sequences using the concepts of domain recurrence to detect related domain sequences in different multidomain contexts. The forthcoming EyeSite database and other resources being developed within the MRC Cooperative will contain families derived using a similar approach based on MKDOM. Both VIDA and EyeSite contain a small proportion of domain families identified structurally by mapping to families in the CATH database.

Whole genome data is currently being maintained within the Gene3D database (Buchan *et al.*, 2001). This identifies CATH protein domain families within 36 completed genomes using a PSI-BLAST based protocol (DomainFinder, Pearl *et al.*, 2001; DRange, Buchan *et al.*, 2001). Sequence relatives for families in Gene3D are derived using a less stringent domain-boundary prediction protocol than that used to derive sequence domains for the CATH database itself.

There may be several conflicting definitions of a protein domain for a particular amino-acid sequence—the CATH definition, the Gene3D definition and/or the VIDA or EyeSite definition. Allowing users to assess the similarities and the differences between these different ways of defining protein domains is one of the core functions of the PFDB.

Significant future developments of the PFDB have already been planned or anticipated. A collaboration with the Macromolecular Structure Database (MSD, Keller *et al.*, 1998) at the European Bioinformatics Institute (EBI) is already underway and will lead in due course to the incorporation within the PFDB of extensive, high-quality data derived from the Protein Data Bank (PDB, Berman *et al.*, 2000). It is also intended that the PFDB will be extended to handle microarray data (notably the virus expression data generated by Paul Kellam's laboratory, Albà *et al.*, 2001) and metabolic pathway data.

THE PFDB DATABASE

PFDB data sources

The PFDB integrates data from a variety of different sources. The starting point is information about the protein classification schemes of the various databases (CATH, VIDA, MRC Cooperative databases). These databases provide descriptions of protein domain families together with the boundary definitions of individual domains within each family. In CATH, domains may be discontiguous with respect to the underlying amino-acid sequence and future releases will contain multi-chain domains, i.e. domains that span more than one chain of amino acids within a multi-chain protein.

The amino-acid sequence data relevant to the CATH and VIDA domains are extracted from the GenBank flat file of non-redundant proteins. For a single sequence in the GenBank non-redundant file, multiple entries are loaded into the database whenever that sequence relates to the separate chains of a protein in the PDB, or when it is attributed to more than one source organism. This information, together with any synonym identifiers (GIs, SWISS-PROT codes) for a given GenBank sequence, is extracted from the concatenated header information that precedes each sequence in the GenBank file.

The source organism information extracted from the GenBank file of non-redundant proteins is mapped into the preferred taxonomic names (both common and scientific) specified by the NCBI taxonomy database (Wheeler *et al.*, 2000; Benson *et al.*, 2000). For entries in the GenBank file that ultimately derive from SWISS-PROT (Bairoch and Apweiler, 2000), this mapping is achieved using the SWISS-PROT speclist file. For entries that ultimately derive from the PDB (Berman *et al.*, 2000), the mapping is achieved using a copy of the NCBI's PDBeast table.

Functional annotations for the various sequences in the PFDB are currently derived from a variety of publicly available resources. For example, SWISS-PROT keywords (Bairoch and Apweiler, 2000) and EC (Enzyme Classification) numbers (Bielka *et al.*, 1992). *E. coli* is one of the most widely annotated genomes, and this data is stored in the EcoCyc (Karp *et al.*, 2000) and GenProtEC (Riley and Serres, 2000) databases. Functional data from GenProtEC can be readily extracted and is currently captured in the PFDB. Future collaborations with Pfam (Bateman *et al.*, 2000) and InterPro (Apweiler *et al.*, 2001) providing mappings between these resources and CATH will create additional functional annotations, as will links to the Gene Ontology (GO) (which includes broader functional descriptions, e.g. cellular location, phenotype as well as biochemical data) that is being modelled by The Gene Ontology Consortium (2001) using data from the yeast, worm and fly genomes.

Information about each of the structures in the Protein Data Bank is also stored within the PFDB. Currently only a subset of the available information about a given PDB structure is stored, information that is extracted from the secondary data source PDBsum (Laskowski, 2001). However, the aim is to incorporate substantially more information as soon as clean PDB data becomes available via the Macromolecular Structure Database (MSD; Keller *et al.*, 1998). In addition, how the structural sequences from the PDB (which omit residues that were not resolved in the relevant crystal structure) map into their corresponding GenBank amino-acid sequences is calculated (using a sequence alignment protocol developed by Lee, personal communication) and stored outside the database, though accessible as table data using SQL functions (Reinwald and Pirahesh, 1998).

In addition to protein sequence information, the PFDB stores information about whole genomes, viral genome fragments (the latter being relevant to the VIDA viral database) and the constituent genes within these genomes. All of this data is extracted from the relevant GenBank genome files. For protein-coding genes, links to the relevant amino-acid sequence data are made by matching the GI identifiers in the GenBank genome files to those in the GenBank flat file of non-redundant proteins. The explicit mappings between genome and gene sequences are extracted from the GenBank genome files.

The interrelationship between the various data sources used by the PFDB is shown schematically in Figure 1.

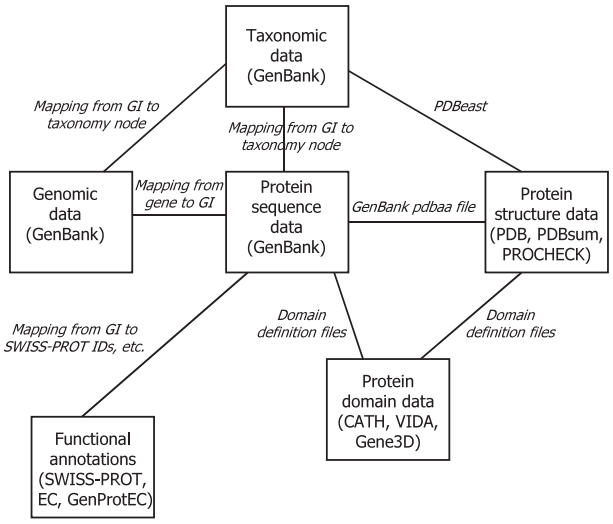

Fig. 1. Schematic diagram of PFDB data sources.

Implementing the PFDB

When implementing the PFDB, we were faced with a fundamental choice—whether to implement it as a relational database, or as an object database. A pilot project, which implemented a subset of the PFDB data using the object DBMS O2 (Deux, 1991), failed to achieve the level of performance required for the PFDB. Further, system support for management and maintenance of a database of reasonable size and complexity were found to be inadequate.

We have, therefore, opted for the mature relational database technology of Oracle 8i. This not only meets our performance requirements, it also has the added advantage that the same RDBMS is being used to manage data at the EBI—in particular the MSD database (Keller *et al.*, 1998), which will in due course become a primary source of data for the PFDB.

One of the key features of the PFDB is the way relational tables are used to represent abstract generic entities. The Unit and Association tables are used to represent binary relationships between biological objects of interest. These tables support a graph representation of the database. The Inclusion, Collection and Aggregation tables represent relationships with additional semantics: set–subset relationships in the case of Inclusion; set membership in the case of Collection; part-whole in the case of Aggregation. The explicit representation of relationships at the meta-level has a number of advantages:

- it makes it possible to ask (meta-level) queries that encompass disparate biological entities. For example, the PFDB *unit* table includes entries for the following biological entities: whole protein structures; protein chains; protein domains; protein domain segments; genes; genomes; each CATH Class, Architecture, Topology, as well as each VIDA domain family;
- it provides support for managing semi-structured data (Buneman, 1997) about biological entities, since the tables directly represent a general graph structure;

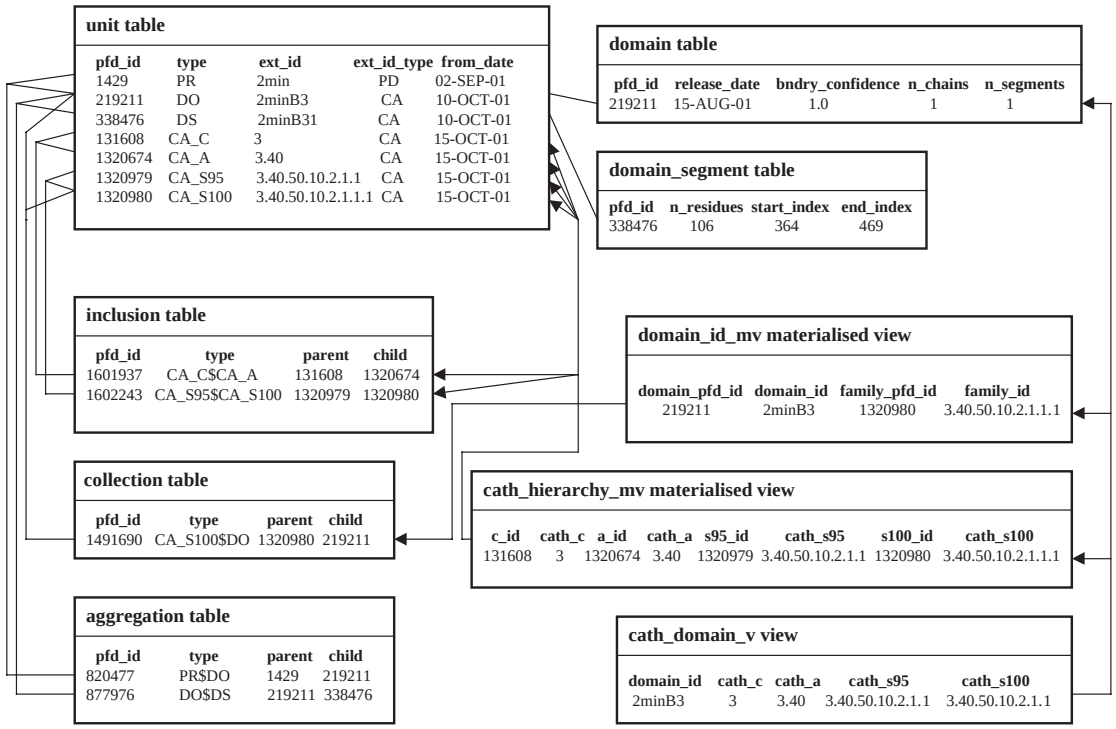

Fig. 2. Schematic diagram illustrating the interrelationship between the base tables and views within the PFDB for example domain 2minB3.

- the meta-level tables can be used to store additional information about a relationship. Two prime examples are: information about the period for which a relationship is valid, which makes it possible to support historical 'versions' of the PFDB; and the degree of certainty with which a relationship is believed to be true, such as the position of domain boundaries;

- it facilitates the future updating of the PFDB by providing a framework for the introduction of new entities.

Built on top of the meta-level and other base tables are a set of materialized (i.e. precompiled, static) views that bring together (denormalized) data from the underlying tables—notably the internal and external identifiers—for improved performance by precomputing results. Finally, on top of the materialized views are a set of standard, dynamic views which present all the relevant attribute information about a particular entity that a typical user is likely to require.

To illustrate how the design works in practice, let us consider a single example—how data about CATH domain 2minB3 is handled in terms of the base tables, meta-level tables, materialized views and standard views. 2minB3 is the domain's external identifier consisting of a PDB code (2min), a chain identifier (B) and a domain number (3). The domain is classified in CATH as 3.40.50.10.2.1.1, which means that 2minB3 belongs to *class* alpha-beta (3), has a three-layer (aba) sandwich *architecture* (3.40), is an example of a Rossmann fold (*topology* 3.40.50) and has been assigned to an *homologous superfamily* associated with nitrogen fixation (3.40.50.10). A schematic diagram of this example is given in Figure 2. Non-arrow lines between base tables indicate foreign key relationships between rows, while arrow lines from views or materialized views to other tables indicate rows which are referenced in a view or materialized view. For clarity, some tables referenced by *domain_id_mv* have been omitted as well as a number of table columns. Also, information about segments that make up a domain is ignored in the following analysis.

Base tables Five base tables—four of which are meta-tables—are used to store protein domain-related information in the PFDB:

- the *domain* table stores some basic attribute information about the domain, notably: *n_segments* (the number of segments that the domain has); and *n_chains* (the number of chains in a given domain). Note that the *domain* table is *not* used to store information about exter-

nal identifiers, domain classifications, or relationships between domains and other structural entities;

- the *unit* table stores the external identifier (2minB3) external label (3—i.e. the domain number), the type of unit ('DO' for domain), the external identifier type ('CA' for CATH), the version, and the dates between which the domain is valid. Table *unit* also has an entry for each node in the CATH hierarchy. For example, there is a unit entry of type 'CA_T' (for CATH topology) for the topology 3.40.50;

- the *inclusion* tables stores the relationships between adjacent levels in the classification hierarchy, for example that between the parent CATH class (3) and the child CATH architecture (3.40). Internal, rather than external, identifiers are used in this table together with some basic typing information (in this case 'CA_C$CA_A' for a CATH class/architecture relationship);

- the *collection* table stores the relationship between a particular domain (2minB3) and its classification (3.40.50.10.2.1.1), the type of classification ('CA' for CATH), the degree of confidence in the classification, and the dates between which the domain classification is valid;

- the *aggregation* table is used to store information about the relationship between a domain and its parent protein (PDB 2min), and between a domain and its child segment(s). These relationships are known as a 'PR$DO'—protein/domain—and 'DO$DS'—domain/segment—relationship respectively. The degree of confidence in the association and the dates between which it is valid are also stored;

- the *association* table is used to record the relationship between a domain and the method used to detect its boundaries.

Since there exist both domains that consist of discontiguous segments of a protein chain (multi-segment domains) and domains that span multiple amino-acid chains within a single protein (multi-chain domains), there is a many-to-many relationship between amino-acid sequence and CATH/VIDA domain. This many-to-many relationship is naturally represented by two 1-many relationships in the aggregation table.

Views There are two materialized views used to store protein domain-related information in the PFDB:

- materialized view *domain_id_mv* maps the external identifiers for a domain—its protein identifier (2min), chain identifier (B), domain identifier (3) and its family identifier (3.40.50.10.2.1.1)—to their corresponding internal identifiers. The information is drawn from the *unit*, *aggregation*, *collection* meta-tables;

- materialized view *cath_hierarchy_mv* maps the external identifiers from table *unit* for a particular CATH classification (e.g. 3, 40, 50, 10, etc.) into their corresponding internal identifiers from table *inclusion*. This materialized view effectively performs a join across fifteen tables.

Finally, a single dynamic view draws together all the relevant information about a CATH domain (excluding the internal identifiers): View *cath_domain_v* combines the external identifiers for individual domains (from *domain_id_mv*) and for the CATH classification (from *cath_hierarchy_mv*) together with the attributes stored in the *domain* table.

The PFDB interface

A Web interface to the PFDB is currently being developed using the Perl DBI. A number of preformulated queries (i.e. parameterized queries with a set format) have already been written based on the needs identified by the developers of the CATH, VIDA and EyeSite databases, and on a requirements analysis carried out within the Biomolecular Structure and Modelling Unit at UCL. Questions that can be asked via the PFDB preformulated query interface (URL http://bsmsn01.biochem.ucl.ac.uk/) include the following:

- what are the PDB codes and GenBank identifiers of sequences belonging to a particular CATH family?

- what products are associated with a particular CATH fold or family?

- what CATH folds or sequence families occur in kingdom *x* but not in other kingdoms?

- what genomes contain at least one example of a particular CATH fold or family?

Taking as an example query 'what genomes contain a particular CATH fold, namely 3.20.20 (TIM barrels)?', the corresponding SQL query references 3 materialized views (each twice), 2 meta level tables and 1 ordinary base table. The response time for these 4 queries ranges between 2 and 10 seconds.

Accessing the PFDB via preformulated queries has several advantages. Preformulated queries are easy to use, they can be highly optimized to guarantee fast response times, and they prevent users from running queries that are inefficient and/or require excessive amount of CPU time. However, preformulated queries do not offer the kind of flexibility that many users desire. It is planned,

therefore, that a flexible interface—one which allows users to compose *ad hoc* queries, but does not require a knowledge of SQL—will be developed in the near future.

DISCUSSION

The PFDB is a new resource that aims to provide fast and flexible access to protein family-related data. It is being used to integrate and manage data for several protein-related databases—the CATH, VIDA, Gene3D and EyeSite databases.

The use of meta level entities in the design supports a flexible framework for adding new entities into the database, and it provides a mechanism for answering complex queries about disparate biological entities.

Given the rapidity with which the bioinformatics field is developing, the ability of the PFDB to integrate new entities and relationships into the existing design with relative ease is of paramount importance. New types of data that we intend to introduce into the PFDB in the next 12 months includes: protein–protein interactions, metabolic pathways, transcriptomic data and proteomic data.

We have already experienced the benefits of our flexible, meta-level approach in our ongoing work aimed at reconciling the PFDB schema with that developed independently for the MSD database (Keller *et al.*, 1998). MSD models structure in much greater detail than the PFDB, reflecting its concern with the detailed crystallographic structure of proteins down to the atomic level (details that lie outside the scope of the PFDB). The changes that need to be made to the PFDB schema in order to establish explicit, well-defined relationships to entities in MSD are negligible, being confined to a small number of base tables. Apart from incorporating some additional typing information, no changes to the underlying meta-tables are required.

The absence of atomic-level data from the PFDB points to another of its key characteristics. Rather than attempt to be comprehensive, the PFDB is by design selective in the data it allows users to search on, preferring high-level information to vast quantities of low-level information (such as atomic-level data). This selectivity has clear performance benefits.

We believe the design decisions taken in the construction of the PFDB have proved to be sound. The combination of meta-tables and conventional relational base tables has given us the modelling power and flexibility of a graph or binary-relational database system without the performance penalties often incurred with such systems when all data is mapped to a graph format. Similarly, we have avoided the performance limitations and weak system management facilities of many current object-oriented database systems. We believe the approach will naturally generalize to incorporate the more general graph structures of metabolic pathways and other transcriptomic and proteomic data.

ACKNOWLEDGEMENTS

This work was funded by the BBSRC/EPSRC Bioinformatics Initiative.

REFERENCES

Albà,M.M., Lee,D., Pearl,F.M.G., Shepherd,A.J., Martin,N.J., Orengo,C.A. and Kellam,P. (2001) VIDA: a virus database system for the organisation of virus genome open reading frames. *Nucleic Acids Res.*, **29**, 133–136.

Altschul,S.F., Madden,T.L., Schäffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Apweiler,R., Attwood,T.K., Bairoch,A., Bateman,A., Birney,E., Biswas,M., Bucher,P., Cerutti,L., Corpet,F., Croning,M.D.R. *et al.* (2001) The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Res.*, **29**, 37–40.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.*, **2**, 45–48.

Bateman,A., Birney,E., Durbin,R., Eddy,S.R., Howe,K.L. and Sonnhammer,E.L.L. (2000) The Pfam Protein Families Database. *Nucleic Acids Res.*, **28**, 263–266.

Benson,D.A., Karsch-Mizrachi,I., Lipman,D.J., Ostell,J., Rapp,B.A. and Wheeler,D.L. (2000) GenBank. *Nucleic Acids Res.*, **28**, 15–18.

Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.

Bielka,H., Dixon,H.B.F., Karlson,P., Liebecq,C., Sharon,N., Van Lenten,E.J., Velick,S.F., Vliegenthart,J.F.G. and Webb,E.C. (1992) *E. C. Enzyme nomenclature 1992: recommendations of the Nomenclature Committee of the International Union Of Biochemistry and Molecular Biology on the nomenclature and classification of enzymes*, Nomenclature Committee of the International Union of Biochemistry, Academic Press, London.

Buchan,D.W.A., Shepherd,A.J., Lee,D., Pearl,F.M., Rison,S.C.G., Thornton,J.M. and Orengo,C.A. (2001) Gene3D: structural assignment for whole genes and genomes in the CATH database. *Genome Res.*, **12**, 503–514.

Buneman,P. (1997) Semistructured data. In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. pp. 117–121.

Corpet,F. (1988) Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res.*, **16**, 10881–10890.

Deux,O. (1991) The O2 system. *CACM*, **34**, 34–48.

The Gene Ontology Consortium (2001) Creating the gene ontology resource: design and implementation. *Genome Res.*, **11**, 1425–1433.

Harrison,A., Pearl,F.M., Sillitoe,I., Slidel,T., Mott,R., Thornton,J.M. and Orengo,C.A. (2002) Graphical representations of architecture, topology and homology. Submitted for publication.

Jones,S., Stewart,M., Michie,A., Swindells,M.B., Orengo,C.A. and Thornton,J.M. (1998) Domain assignment for protein structures using a consensus approach: characterisation and analysis. *Protein Sci.*, **7**, 233–242.

Karp,P.D., Riley,M., Saier,M., Paulsen,I.T., Paley,S. and Pellegrini-Toole,A. (2000) EcoCyc: electronic encyclopedia of E. coli genes and metabolism. *Nucleic Acids Res.*, **28**, 56.

Karplus,K., Barrett,C. and Hughey,R. (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, **14**, 846–856.

Keller,P.A., Henrick,K., McNeil,P., Moodie,S. and Barton,G.J. (1998) Deposition of macromolecular structures. *Acta Crystollogr.*, **D54**, 1105–1108.

Laskowski,R.A. (2001) PDBsum: summaries and analyses of PDB structures. *Nucleic Acids Res.*, **29**, 221–222.

Orengo,C.A., Michie,A.D., Jones,S., Jones,D.T., Swindells,M.B. and Thornton,J.M. (1997) CATH—a hierarchic classification of protein domain structures. *Structure*, **5**, 1093–1108.

Pearl,F.M.G., Lee,D., Bray,J.E., Sillitoe,I., Todd,A.E., Harrison,A.P., Thornton,J.M. and Orengo,C.A. (2000) Assigning genomic sequences to CATH. *Nucleic Acids Res.*, **28**, 277–282.

Pearl,F.M.G., Martin,N., Bray,J.E., Buchan,D.W.A., Harrison,A.P., Lee,D., Reeves,G.A., Shepherd,A.J., Sillitoe,I., Todd,A.E., Thornton,J.M. and Orengo,C.A. (2001) A rapid classification protocol for the CATH domain database to support structure genomics. *Nucleic Acids Res.*, **29**, 223–227.

Reinwald,B. and Pirahesh,H. (1998) SQL open heterogenous data access. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. Seattle, Washington, pp. 505–507.

Riley,M. and Serres,M.H. (2000) Interim report on genomics of *Escherichia coli*. *Annu. Rev. Microbiol.*, **54**, 341–411.

Schäffer,A.A., Wolf,Y.I., Ponting,C.P., Koonin,E.V., Aravind,L. and Altschul,S.F. (1999) IMPALA: matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices. *Bioinformatics*, **15**, 1000–1101.

Wheeler,D.L., Chappey,C., Lash,A.E., Leipe,D.D., Madden,T.L., Schuler,G.D., Tatusova,T.A. and Rapp,B.A. (2000) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, **28**, 10–14.