# BioMap: Gene Family based Integration of Heterogeneous Biological Databases Using AutoMed Metadata

Michael Maibaum[†*]
*maibaum@biochemistry.ucl.ac.uk*

Galia Rimon[†]
*galia@dcs.bbk.ac.uk*

Christine Orengo[*]
*orengo@biochemistry.ucl.ac.uk*

Nigel Martin[†]
*nigel@dcs.bbk.ac.uk*

Alexandra Poulovassilis[†]
*ap@dcs.bbk.ac.uk*

## Abstract

*This paper presents an extensible architecture that can be used to support the integration of biological data sets. Biological research frequently requires this kind of synthesis. However, the data models on which biological data sets have been constructed are heterogeneous and difficult to use together. Our architecture uses the AutoMed data integration toolkit to store the schemas of data sources, together with the transformation from these schemas into a global integrated schema.*

*The transformation encompasses two parts; the incremental construction of a global schema which unifies the various data source schemas, and the identification of semantically identical labels for entities.*

*Entities in the unified resource are integrated using PFScape. This categorises the entities into clusters based on sequence similarity, allowing the use of family information in the annotation of expression data and experimental target selection.*

## 1. Introduction

The integration of multiple large, diverse biological data sets is a daunting problem. There are three major obstacles. The first is the use of different identifiers for the same entities, the second the diversity of data models underpinning the biological data and finally the requirement to keep an integrated resource current.

Even within a single data source different identifiers are in use for the same entity. For example the Gene Ontology refers to the Mouse Genome database as MGI, MGD and 'MGI (presumably a data entry error). Furthermore there is widespread use of inappropriate identifiers such as NCBI GI numbers. GI numbers identify a submission to GenBank. If the submission is changed in any way a new GI is issued. As many people may sequence the same gene or peptide there are many GI numbers for a single entity. The GI cannot be used to determine which sequences in GenBank are the same.

A vast range of different identifiers exist for the same or similar biological entities, so even if two data sources are perfectly maintained they may well be impossible to map together based on the identifiers in use if they have chosen a different set of reference identifiers to use.

The integration of facts about biological entities presented here is based on sequence similarity. Not only does this approach avoid the problematic use of identifiers completely, the use of metadata about how entities are related allows us to mine the information available about each family and not be limited to individual biological entities.

Each source of data has its own structure that is a consequence of the domain (biological focus) of the resource and the structure in which it is stored. This heterogeneity of structure makes it extremely difficult to design a generalised interface to multiple data sources and to integrate the information within those data sources. To be able to present a unified view of the facts in the various resources requires the provision of an abstract interface to the underlying data stores.

Our architecture uses an AutoMed metadata repository to specify how each particular data structure can be transformed into an abstract form. AutoMed is a data transformation and integration framework which can be used for both virtual and materialised data integration (see http://www.doc.ic.ac.uk/automed for a list of technical reports and papers relating to AutoMed). It has a low-level hypergraph-based data model (the HDM) as its internal data model, and provides facilities for defining higher-level data models, e.g. the relational data model, XML and flat file formats, in terms of this lower-level HDM. AutoMed provides transformation primitives that can be used to transform one schema into another, or to integrate a number of data source schemas into a global schema. Schemas are incrementally transformed by applying to them a sequence of primitive transformations

[†]School of Computer Science and Information Systems,
Birkbeck College, London
[*]Department of Biochemistry,
University College London

$t_1 ... t_r$ each of which changes the current schema by adding, deleting or renaming just one schema construct. Each **add** or **delete** transformation is accompanied by a query specifying the extent of the new or deleted construct in terms of the rest of the constructs in the schema. These queries make possible automatic query and data translation between schemas linked by transformation pathways.

AutoMed metadata has enough expressivity to describe the data cleansing, integration and transformation processes of a data warehouse and this metadata can be used in warehouse activities such as populating the warehouse, incrementally maintaining it after data source updates, and tracing the lineage of the data within it [1] .

The architecture presented here is currently being applied in the construction of the BioMap data warehouse (see http://www.biochem.ucl.ac.uk/bsm/biomap). This warehouse incorporates protein family, structure, function, and pathway/process data, together with gene expression and other experimental data. Data mining and associated visualisation tools are being developed within the BioMap project which exploit the integration of expression data and knowledge of protein families and functions within the warehouse in order to support the analysis of co-expressed genes and to drive experimental design (e.g. knowledge based target selection for structural genomics programs). BioMap data sources currently include the EBI Macromolecular Structure Database (EBI-MSD) (http://www.ebi.ac.uk/msd [2]), CATH (http://www.biochem.ucl.ac.uk/bsm/cath [3]), KEGG (http://www.genome.ad.jp/kegg/), InterPro (http://www.ebi.ac.uk/interpro/), Gene Ontology (http://www.geneontology.org/) and 10 other resources.

## 2. The System Architecture

Figure 1 shows the main components and processes of our data integration system.

Each data source has its own schema (A). There are two kinds of source schema, which are processed in the same way but contain different kinds of information. A **data source schema** is a source of information about a subset of entities in the warehouse as a whole and

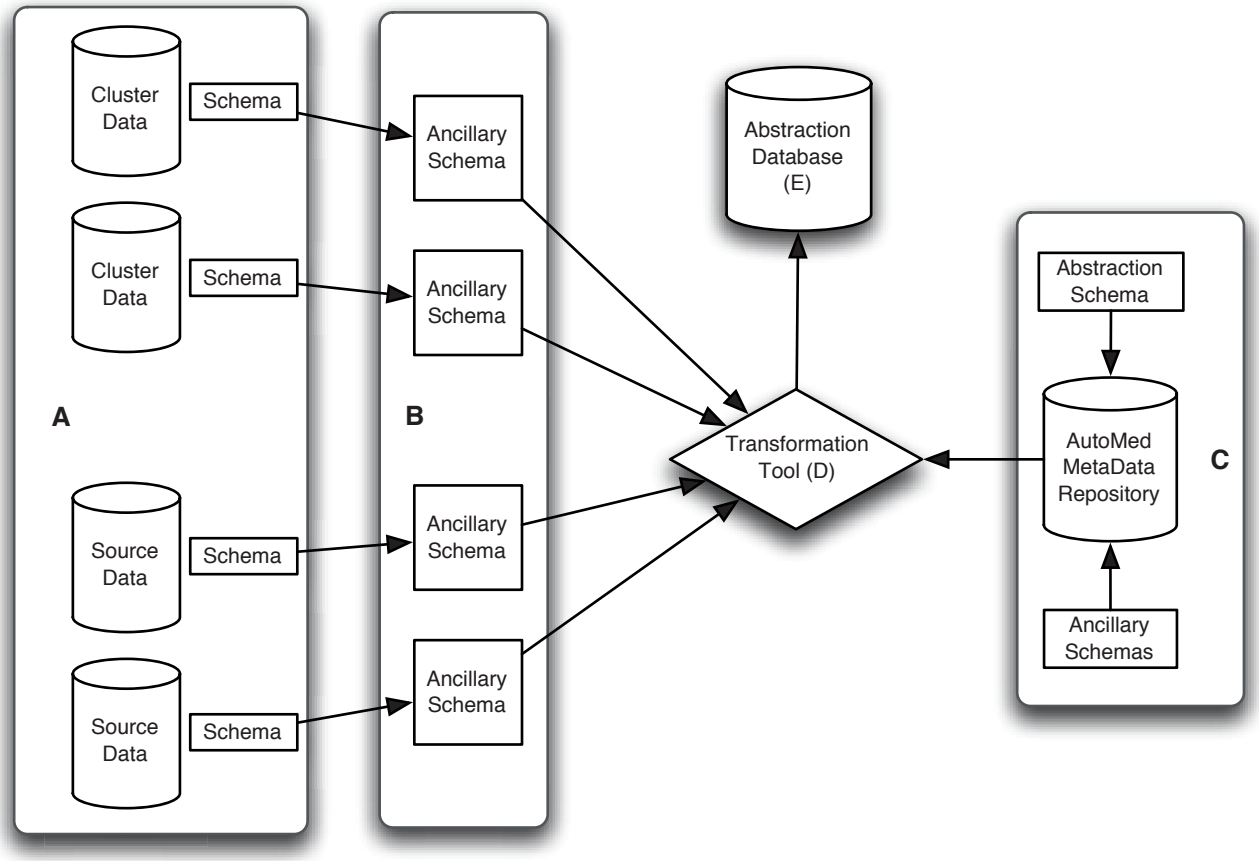


**Figure 1. Overview of the System Architecture.**

describes facts about those entities. Examples of this type of source schema are CATH or KEGG. A **cluster schema** contains information about how entities in the data sources relate to each other and can be used to create groups of entities that are logically related in some way (e.g. sequence similarity). These schemas will usually have information about all the entities in the data sources.

An **ancillary schema** for each source schema (B) is constructed from the original source (A). The ancillary schema provides a persistent store of the entities from each source schema providing a non-volatile identifier for each entitiy. This provides the information needed to keep the rest of the warehouse in sync with the changing content of the sources.

An **AutoMed repository** (C) stores metadata which describes the relationships between the entities within a specific source/ancillary schema. This metadata is used by a **transformation tool** (D) to transform the data in the source/ancillary schemas into the schema of the **abstraction database** (E) and store it there. This abstraction database is currently a relational database. Data can then be used to construct tailored views and data marts through which users can access the data. The abstraction database also allows the comparison of entities between different versions of a source: once changes between between different versions of a source have been identified, they can be applied to the data in the abstraction database. The changes can then be propagated from there to the views and data marts.

## 3. Schemas, Data and Metadata

### 3.1 Source Schemas and Data

The individual data sources contain a diverse set of topics, data structures, formatting conventions and sizes. The resources describe structural, functional, sequence and ontological information. At least one source schema also contains the data required for the sequence family based integration of the sources.

The storage structures vary from conventional relational structures, XML, to various flat file formats. Internally each resource uses different formatting conventions so semantically identical cross-references between databases may not match in a string comparison. For example the GO term 'GO:12345' could be stored as '12345'. The resources can also be very large. The EBI Macro-Molecular Structures Database requires over 150Gb of storage space.

### 3.2 Ancillary Schema

In many cases the schema of the original data source does not provide a simple identifier for each entity within the data set and a composite key is required to be able to reliably identify a specific entity. Without a reliable identifier for entities in a data source, the identification of changes between versions of the data source is extremely difficult. The ancillary tables provide these stable identifiers. Tables 1-3 show an example set of tables in the ancillary schema for the KEGG Orthology. Table 1 contains the principal 'Orthology' entity along with a set of facts pertaining to the entity. Table 2 defines the Gene entities that the orthology contains. Table 3 contains cross references to entities from other data sources.

Only the facts that pertain to entity identification are

**Table 1. Orthology**

| ID | Entry | EC Number | Symbol | Name |
|---|---|---|---|---|
| 1 | K00001 | 1.1.1.1 | adh | alcohol dehydrogenase |
| 2 | K00002 | 1.1.1.2 | adh | alcohol dehydrogenase (NADP+) |
| 3 | K00003 | 1.1.1.3 | thrA | homoserine dehydrogenase |
| 4 | K00004 | 1.1.1.4 | | (R,R)-butanediol dehydrogenase |
| 5 | K00005 | 1.1.1.6 | gldA | glycerol dehydrogenase |

**Table 2. Gene**

| ID | Entry | Species | Gene ID | Symbol |
|---|---|---|---|---|
| 1 | K00001 | STY | STY1493 | adhP |
| 2 | K00001 | STY | STY3830 | |
| 3 | K00001 | STY | STY1302 | adh |
| 4 | K00001 | SAM | MW0568 | adh1 |
| 5 | K00001 | SAM | MW0123 | adhE |

**Table 3. Cross References**

| ID | Entry | Xref DB | XRef ID |
|---|---|---|---|
| 1 | K00001 | EC | 1.1.1.1 |
| 2 | K00001 | GO | 0004022 |
| 3 | K00001 | GO | 0004023 |
| 4 | K00001 | COG | COG1012 |
| 5 | K00001 | COG | COG1062 |
| 6 | K00002 | EC | 1.1.1.2 |

included in the ancillary tables, namely internal and external identifiers of entities and classifications of those entities. Remaining information about a specific entity can be obtained by querying the source directly.

The ancillary schemas provide an additional benefit as they present a simplified subset and relational view of the source data. There is no technical reason that the abstraction could not be derived directly from the original source schema assuming that it provides persistent identifiers. However using these ancillary schemas eases the task of describing the relationships within the data source and the transformation of the facts into an abstract form.

### 3.3 Metadata Repository

The metadata repository stores information about each of the above schemas and the relationships between them. This information is used by the transformation tool to transform the source/ancillary data into the form required by the abstraction schema, and also to identify changes between versions of a data source.

Two types of information are required for the transformation of the source data. The first is a definition of the relationships between the entities within a data source. For example, tables 1-3 contain instances of the relationships shown in Figure 2. We see that an Orthology entity consists of an Entry (KEGG identifier), an EC number, a symbol, a set of Gene Entities and a set of cross references to external resources. The Gene Entities themselves have an Entry, a symbol, a gene ID and a species. The metadata repository describes this set of relationships and how they map onto the tables of the

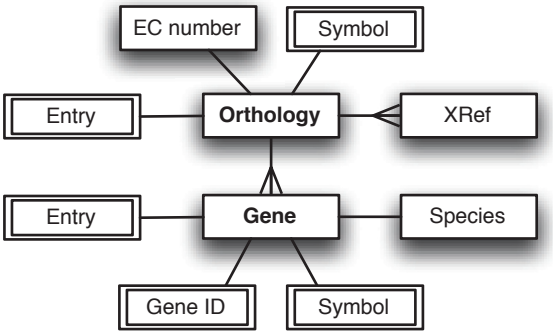


**Figure 2. Entity Relationship diagram for Entities in Tables 1-3.**

abstraction schema (see below).

The metadata repository also stores information on standardisation of identifier formatting. For the abstraction database to be as useful as possible it must store identifiers that are used in multiple resources in a standardised fashion. Using the example of a GO term as above; if the standardised form for a GO Term has been defined as the string 'GO:' followed by a string of digits then any GO Terms stored as a string of digits should have the 'GO:' string prepended. AutoMed is used to store a transformation from one format to the other in the same way it stores the information required for the transformation of schema structures.

### 3.4 Abstraction Schema

The abstraction schema consists of a table containing each fact, and one or more mapping tables that encapsulate the relationships between facts, as defined in the metadata repository.

The fact table contains the source table for each fact (schema and table name) and the primary key for that table. This allows the precise identification of a fact in the source table within the ancillary schema. Each fact is typed and the value of the fact and its type stored (after it has been correctly formatted based on the information stored in the AutoMed repository). The fact table is sufficient to store all the facts from a source schema but does not allow you to correctly connect the facts into the networks as described by Figure 2.

The connections between the facts form a graph that is described in additional tables [4] and can be used to extract the related entities from the fact table. There is no inherent limitation to the type of mapping data stored. The mapping between the entities may be a simple binary graph describing parent-child relationship. Complex graph objects are also supported. More complex graphs provide a basis to represent the emergent networks between the biological entities in the various data sources.

## 4. Maintenance

A data warehouse must be kept up to date to be useful. The abstraction database provides a persistent store of the entities within each resource in a standardised format. When a resource is updated, the data from that source can be transformed into the form of the abstraction schema, and can then be compared to the existing facts in the abstraction databasse. In this way, entities that have changed, been added, or been deleted can be identified and updated in the abstraction database. Once the abstraction database has been brought up to date, the changes can then propagate to the various views and data marts derived from it.

The abstraction database allows changes in the source schemas to be isolated from applications. This

method also allows the addition or subtraction of entire resources, for example a new biological database could be added and simply combined with the data already present within the abstraction database. Once such additions, modifications or deletions have taken place, applications can be modified as needed to take advantage of any extra data. Such modifications will be simple as the abstraction schema upon which they are based will not have changed.

## 5. Integration

The abstraction of the facts from the individual resources supports a standardised interface to the facts in each member data set. The standardisation of identifiers allows an improvement in the integration of the data sets. However the overlap based on identifiers is relatively low, below 40% in some cases, even when the data sets are nominally describing the same entities.

There are a variety of methods of classifying biological entities into sets and these methods can be used on the facts within the data warehouse. The facts concerning individual entities within a set will not all derive from precisely the same biological entity, but by choosing an appropriate algorithm to create the sets, the set will contain valuable information about biological entities that are similar (in some way) to each other. One such categorisation method is UniGene (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=unigene). Our categorisation method is based on the PFScape protocol [5] which is in turn based on the TRIBE-MCL algorithm [6].

The use of sequence families to identify clusters or groups of similar or related entities allows a significant improvement in the possible annotation of the individual members of a group. Sequences are assigned to clusters based on levels of sequence similarity (100% to <35%) with facts recorded in a data set also being assigned to these clusters. The approach supports the integration of multiple data sets at a level of similarity appropriate to the type of data being integrated. Protein structure is conserved at low levels of sequence similarity compared to function and therefore clusters with lower levels of similarity can be used when structural annotation is desired rather than functional.

While the use of sequence families is described here, other methods of classification could be used, these include structural and many other approaches. Whichever approach is chosen, the classification of each entity is represented as a data set in the same manner as each of the other data sets. The classification information is thereby included in the abstraction database which can then support queries to return sets of data relevant to particular categories of entities.

## 6. Conclusions

At the present time the integration of biological data sets is extremely difficult. The problem is well recognised within the community and a number of efforts have been initiated to help overcome the problems. Over the next few years the situation will improve as various projects (e.g. UniProt - http://www.ebi.ac.uk/uniprot/) mature and more standard identifiers increasingly become available and are used in more data sources.

Even with the increased use of standard identifiers most of the problems of integrating biological, or any diverse group of data sets, will remain. The ability to abstract the data and treat many resources in a similar manner allows the construction of tools to keep the data on which applications depend up to date, while integrating the data in a flexible and powerful manner. The use of family based data to annotate entities provides valuble information that would otherwise be missed.

The solutions presented in this paper provide an extensible approach to providing an abstract interface to these diverse resources. This abstract interface can then be used to maintain a wide variety of applications, including data marts, specialised applications and web services.

## References


[1] Fan H., Poulovassilis A., “Using AutoMed metadata in data warehousing environments.” DOLAP 2003: 86-93

[2] Golovin A., Oldfield T.J., *et. al.* “E-MSD: an integrated data resource for bioinformatics.” NAR. 2004 Jan 1;32 (Database issue), pp. D211-D216

[3] Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B., and Thornton, J.M. “CATH- A Hierarchic Classification of Protein Domain Structures.” Structure. 1997 5(8) pp. 1093-1108.

[4] Shepherd A. J., Martin N.J., Johnson R.G., Kellam P., Orengo C.A. “PFDB: A generic protein family database integrating the CATH domain structure database with sequence based protein family resources.” Bioinformatics. 2002 18 pp. 1666-1672.

[5] Lee D.A., Fefeu S., Edo-Ukeh A.A., Orengo C.A., Slingsby C. “EyeSite: a semi-automated database of protein families in the eye.” NAR. 2004 Jan 1;32 Database issue pp. D148-52.

[6] Enright A.J., Van Dongen S., Ouzounis C.A. “An efficient algorithm for large-scale detection of protein families.” NAR. 2002 Apr 1;30(7) pp. 1575-84.