COLUMBA: Multidimensional Data Integration of Protein Annotations

Kristian Rother[1], Heiko Müller[2], Silke Trissl[2], Ina Koch[3], Thomas Steinke[4],
Robert Preissner[1], Cornelius Frömmel[1], Ulf Leser[2]

[1] Universitätskrankenhaus Charité Berlin, Institut für Biochemie, Monbijoustr. 2
10098 Berlin, Germany
{kristian.rother, robert.preissner, cornelius.froemmel}@charite.de
[2] Humboldt-Universität zu Berlin, Institut für Informatik, Unter den Linden 6,
10099 Berlin, Germany
{hmueller, trissl, leser}@informatik.hu-berlin.de
[3] Technische Fachhochschule Berlin, Seestr. 64, 13347 Berlin, Germany
ina.koch@tfh-berlin.de
[4] Zuse Institut Berlin, Takustrasse 7, Berlin, Germany
steinke@zib.de

Abstract. We present COLUMBA, an integrated database of protein annotations. COLUMBA is centered around proteins whose structure has been resolved and adds as much annotations as possible to those proteins, describing their properties such as function, sequence, classification, textual description, participation in pathways, etc. Annotations are extracted from seven (soon eleven) external data sources. In this paper we describe the motivation for building COLUMBA, its integrational architecture and the software tools we developed for the integrated data sources and keeping COLUMBA up-to-date. We put special focus on two aspects: First, COLUMBA does not try to remove redundancies and overlaps in data sources, but views each data source as a proper dimension describing a protein. We explain the advantages of this approach compared to a tighter semantic integration as pursued in many other projects. Second, we highlight our current investigations regarding the quality of data in COLUMBA by identification of hot spots of poor data quality.

1 Introduction

In life science research, there is an overwhelming amount of data in public and commercial databases available for data analysis and knowledge discovery. The time and cost effective usage of these data is hampered by two main problems: (i) the distribution of relevant data over many heterogeneous data sources and (ii) the quantity of errors and omissions within these sources. The first problem is solved by data integration approaches, while the second problem is tackled by means of data cleansing.

COLUMBA is a database of integrated protein annotations. Therefore, it has to cope with both types of problems. First, the sources currently integrated into COLUMBA are spread world-wide, are hosted on a variety of different platforms, and each has its own proper schema or format, semantically and syntactically distinct from all others. Second, the sources suffer from incompleteness and sometimes store redundant results, which need to be identified and inconsistencies need to be removed. Within this paper, we explain our specific solutions to both problems.

Data integration in general is complicated by technical, syntactical, and semantic heterogeneity of the sources, our inability to define a global, all-embracing schema for the domain due to incomplete domain knowledge and the pure complexity of the problem, frequent changes to source data and source schemas, and the effort necessary for integrating additional data sources. Data integration systems aim at providing unified access to a set of heterogeneous data sources based on an integrated view of the data sources. For COLUMBA we choose a multidimensional integration approach, where the data are materialized in a local relational database and the global schema is built of linked component schemas for each of the sources, closely following their proper concepts. The schema is centered on protein structures, taken from the central repository, the Protein Data Bank (PDB) [1]. Apart from the PDB, COLUMBA currently integrates six different data sources, describing various aspects of proteins and protein structures.

We call our approach "multidimensional" because it treats each data source as an essentially independent dimension, describing proteins. Keeping the schema close to the concepts of the sources and connecting them by relations has the advantage of (i) presenting all data to the biological user in terms of well-known concepts, which would get blurred by semantic integration, (ii) relieving us of the necessity for semantic integration, (iii) enabling the re-usage of existing parser software for the sources, (iv) effortless addition and removal of sources due to modular design of the schema, (v) simplifying the reconstruction of the origin of each piece of information (data provenance) resulting in higher user confidence and an ease of the update process in case of source changes, and (vi) a simple and intuitive query model (and resulting web-interface) following a 'query refinement' paradigm resembling common manual procedures of the biological domain expert.

Many data sources in the biomedical domain are renowned for containing data of sometimes poor quality [2][3]. This is due to the experimental nature of the field, the quickly changing knowledge landscape, the high redundancies in experiments performed often leading to contradicting results, and the difficulties in properly describing the results of an experiment in a domain as complex as molecular biology. Furthermore, it was often observed that data quality problems multiply when data of low quality are integrated and re-used for annotation [4].

In COLUMBA, we pay special attention to the aspect of measuring data quality and detecting hot-spots of poor quality. We approach the problem by analyzing contradicting values in the case of duplicate protein entries. In COLUMBA, such duplicates do not appear at the data sources, which are considered independent, but in the core data, i.e. the PDB entries, itself. Currently there are three instantiations of the PDB, which are derived from each other, but with different procedures for data completion and correction applied. For COLUMBA we face the problem of having to choose the best origin for each single attribute value.

With this paper we want to share our experiences gained in building an integrated database in the life science domain and highlight the benefits and difficulties of the multidimensional integration approach. Further, we report on preliminary results in identifying and quantifying quality problems in complex domains such as life science. The structure of the paper is as follows. In the next section we give an overview of the biological background and motivation for an integrated database of protein structure annotation. In Section 3 we briefly describe the characteristics and formats of the data sources integrated into COLUMBA. Section 4 describes the multidimensional data integration architecture. In Section 5 we describe the design of the database schema and integration pipeline used to populate and update the database. We also outline the possibilities for initial data cleansing during this process. In Section 6 we describe the web-interface for building and executing queries to the database. Section 7 discusses related work. We conclude in Section 8.

2 Biological Background

Researchers on protein structures are often interested in sets of proteins, sharing certain properties such as sub-folds, protein function, source organism, or pathways. Being able to generate such sets quickly has at least two major applications. First, researchers may select such a set to perform some special analysis only on the structures within this set, trying to find set-specific properties. Second, researchers have defined groups of proteins and try to correlate these groups according to the properties of the protein structures contained.

Sets of structures are mainly required for the prediction of the docking of small molecules, protein folding and protein-protein interactions, and analyzing functional relationships. All three require data from comparative studies, also carried out on specific datasets. Depending on the kind of study and interest of the research group a spectrum of very different questions arise. Examples are:

- Get all structures having the *TIM-barrel fold* with a resolution better than 2.0 Å.
- Get all structures of antibodies of the *IgG class*.
- Get all structures of enzymes in the *fatty acid degradation* pathway.
- Get all structures from *saccharomyces cerevisiae* (*yeast*) with less than 90% sequence similarity to a human protein.

These questions have in common that they result in a list of protein structures. The second kind of analysis, e.g., looking for a new structural motif, requires such a list as input, which is then further characterized. The criteria used for characterizing them are similar as in the above examples, because some biases in the source data have to be omitted.

We give one more elaborate example for the types of queries we pursue with COLUMBA. Proteins act together in pathways. For pathways, there are specialized databases such as KEGG [5]. However, the question is whether all pathways known today are equally well represented in terms of structures of the participating proteins. To get an impression of both, the numbers and the heterogeneity in the database, the coverage of metabolic pathways by structures was analyzed, using the integrated data in COLUMBA from the PDB and KEGG. A great mixture of coverage can be observed among pathways, some highly covered while others have no structure assigned to them at all (Table 1).

Table 1. Coverage of metabolic pathways with structures in PDB. Each pathway - a group of connected biochemical reactions - consists of several enzymes. For each enzyme, zero to many protein structures are available. If no structure for an enzyme exists, this is mainly due to experimental reasons. We analyzed to what extent pathways are already resolved by structures. This kind of query is available as a predefined button in the COLUMBA web interface

Pathway	Enzymes	Structures	Enzymes with 1+ structures
Carbon fixation	23	323	87%
Fatty acid biosynthesis II	7	32	57%
Val/Leu/Ile biosynthesis	15	38	53%
Citrate cycle (TCA cycle)	23	157	52%
Peptidoglycan biosynthesis	16	35	50%
Alkaloid biosynthesis I	34	96	15%
Ubiquinone biosynthesis	10	0	0%

3 Data Sources

COLUMBA currently integrates data from seven data sources: The Protein Data Bank (PDB), the Structural Classification of Proteins (SCOP) [6], the Kyoto Encyclopedia of Genes and Genomes (KEGG), the Dictionary of Protein Secondary Structure (DSSP) [7], the Enzyme Nomenclature Database (ENZYME) [8], the Dictionary of Interfaces in Proteins (DIP) [9], and the Protein Topology Graph Library (PTGL). Four more data sources are already physically integrated, but not yet released into the public version of COLUMBA, i.e., protein classification from the Hierarchic Classification of Protein Domain Structures (CATH) [10], functional annotation integrated from SWISS-PROT [11], the NCBI taxonomy database of organisms [12], and the Gene-Ontology (GO) [13].

Additional data sources, which we are planning to integrate soon, are the non-redundant lists of structures from PISCES [14], protein families from SYSTERS [15], and biochemical pathway charts from Boehringer-Mannheim [16].

The Protein structure database PDB is often referred to as a worst case scenario of biological data representation. We will illustrate five typical problems occurring with the PDB and other biological databases as well: (i) Starting in the 1980's as a repository of a few ASCII files of x-ray protein structures, the PDB has grown to the size of more than 22000 structures by late 2003. However, the file format originally inspired from punch cards has changed only once in those 20 years and is very difficult to parse. (ii) It contains structures not only of proteins, but also DNA, peptide fragments and large biological complexes. (iii) The structures have been resolved with several experimental methods, e.g., NMR ensembles and very low resolution electron microscopy images. The quality of the results therefore varies greatly. (iv) The data are highly redundant; there are more than 800 structures of the protein *Lysozyme*, but only one of *Photosystem I.* (v) The annotation is mostly incomplete, erroneous, and full of typographical errors. Controlled vocabularies are not applied in the database until today (but sometimes in submission tools). Thus, one has to pick relevant information very carefully from a data pool with a certain level of noise.

The data sources are available in various formats. The predominant format are flat files, e.g., for PDB, SCOP, CATH, and ENZYME, which are parsed to create load files for the target relational database management system. Boehringer and KEGG were parsed from web sites using our own parsers. Several other sources are available as database dump files, e.g., SWISS-PROT, GeneOntology, and NCBI data [17]. SWISS-PROT data are also provided as flat files and database load files. There exist public parsers for PDB, SCOP, ENZYME, and SWISS-PROT. The remaining data sources in flat file format follow a predominantly simple file format making implementation of the parser fairly simple. DSSP is a special case, because it is a program to compute the secondary structure for protein sequences from the PDB entries. PDB is also available in different formats and database versions. In Section 5.3 we shall give more information how PDB is parsed and transformed to form the basis of COLUMBA.

We integrate everything within a relational database. At time of writing the COLUMBA database contains annotation for 22453 protein structures, containing a total of 45037 polypeptide chains, 25920 biological compounds and 217015 small molecular groups. Of these entries, 89% have been annotated by at least one type of data source (sequence, function or fold), 36% by all three.

4 Multidimensional Data Integration

Defining a global schema covering semantically and structurally heterogeneous data sources poses a major challenge for the design of any data integration system. The approach most commonly described in literature is schema integration. Schema integration is defined as the activity of integrating the schema of existing data sources into a global schema by unifying the representation of semantically similar information that is represented in a heterogeneous way across the individual sources [18]. This involves the identification of semantically equivalent attributes or concepts within the different sources and the definition of a strategy for merging them into a resulting schema, covering all the aspects of the contributing sources. Recently, a number of tools have been proposed which can aid the user in this task by analyzing names and relationships of schema elements [19][20].

Within COLUMBA we use a different approach, which we call multidimensional data integration. The name is inspired from data warehouse design. In data warehousing, facts, e.g., sales, are described by dimensions, such as store, product, or customer [21]. The resulting schema is called star or snowflake schema in correspondence with the visual appearance. In analogy we center our schema on protein structures and describe them by dimensions such as function, description, classification, or sequence. Furthermore, we strictly follow the design paradigm that each data source is mapped into one dimension. Data from different sources are never mixed with each other.

The two main characteristics of our multidimensional data integration approach are (more details are given below):

- We leave the data in a schema closely following the concepts and design of the source schema.

- We model COLUMBA as a multidimensional schema, where the integrated single sources take the role of (often complex and hierarchically structured) dimensions to form a snowflake-like schema centered on the PDB entry.

4.1 System Architecture

The integration process as shown in Fig. 1 is divided into horizontal and vertical integration flow. In the first step for each of the sources a horizontal integration module performs the transformation of the original data into an initial relational representation (initial schema). The complexity of this step depends on the data source. The relational representation filled in this step is completely independent from the final COLUMBA schema.

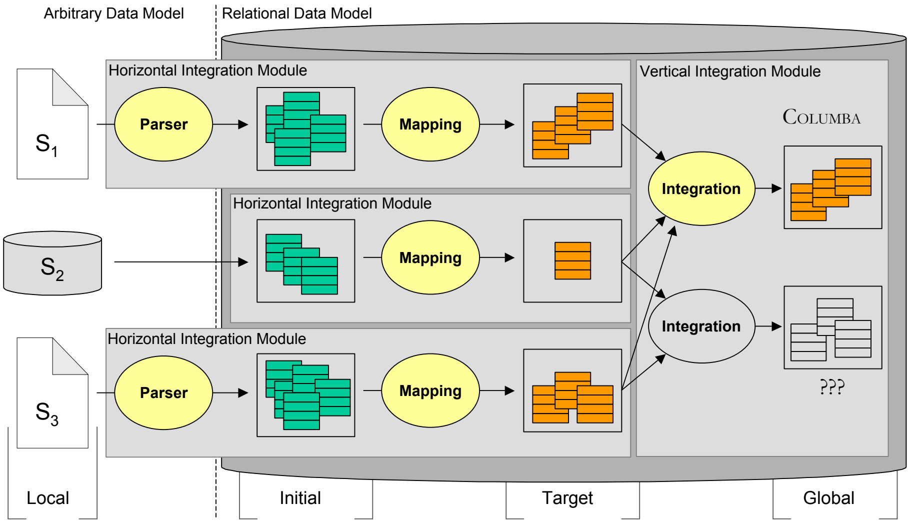


Fig. 1. Multidimensional data integration architecture showing the horizontal integration modules for each of the sources transforming the data from local schema to initial schema and target schema and the vertical integration module integrating the data within the global schema

In the second step, we define a mapping from the initial relational schema into the COLUMBA target schema. Again, this step is performed for each source, i.e., both mappings and the target schema are source-specific. The target schema can differ from the initial schema for many reasons. First, only data relevant for COLUMBA are taken into the target schema. Second, the target schema is modeled such that typical queries can be computed with sufficient performance, e.g., it is often more or less normalized than the initial schema. Third, during the mapping, sometimes also records (and not only tables) are filtered because they describe data items outside the scope of COLUMBA.

In the third step, vertical integration is performed by connecting the data in the target schemas to each other and to the PDB core by filling the linkage tables. This can be done either by using cross-link information from the sources or by performing computations built on knowledge of biological rules. In Section 5 we give a more detailed overview about the actual integration process of COLUMBA.

Having sources within the relational model enables us to define other global schemas as composition of the different sources, depending on the application requirements. We view COLUMBA as only one of the possible target applications.

4.2 Advantages of Multidimensional Data Integration

Multidimensional data integration has advantages regarding system maintenance and development, data provenance, keeping the data up-to-date, and usability. In the following sub-sections we give a detailed overview of the advantages within each of the mentioned areas.

Please note that not performing semantic integration of data sources also has a drawback. For instance, functional annotation of proteins in COLUMBA can be found both in SWISS-PROT description lines as well as in GO annotations. However, we strongly believe that integrating such data is counter-productive. Biologists heavily depend on the credibility of a source and the specific process-of-generation for judging data from databases. Therefore, our users have a clear separation between functional annotation in SWISS-PROT (developed and maintained in human language by a human expert) and functional annotation in GO (also maintained by humans, but in the form of IDs referencing external keywords). Integrating such information would not only be a tremendous and extremely expensive task, but would also blur this vital difference and thus leave users with less confidence to the system.

4.2.1 System Maintenance and Development

Focusing on individual data sources and the distinction between horizontal and vertical integration flows results in a modular system, benefiting extensibility and software development. Adding additional sources or removing existing ones is easy because in case of additions we extend the global schema by including the target schema of the new source as an additional dimension and in case of removal just delete the target schema and the linking tables from the global schema. The modular design also reduces the effort necessary for reacting on changes in the schemas of individual sources. Here again, we only take the individual source into account, define mappings and linkages for them, while the rest of the schema and database remains untouched.

The strong focus on individual data sources also has an advantage in creating the system software for loading and updating the database because in many cases we are able to apply existing software for parsing and loading (see Section 3). Specifying and implementing software parsers is also much simpler if we do not have to take into account any semantic mappings and global constraints. This accelerates the software development part and keeps the resulting system maintainable and well structured.

4.2.2 Data Provenance and Currency

Integrating the data in primarily separate parts for each source facilitates the determination of data provenance, i.e., reconstructing for each piece of information the source and data item it originates. We store information about data provenance and the data integration process as meta-data within separated tables in the global schema. This includes the original data source, the release number, and the programs used to extract or calculate the attribute values for each entry.

Detailed information about the data source of origin for an entry has advantages in case of error identification and source updates. If we identify a specific source entry to be erroneous we can correct it or simply exclude it from our database. In case of changes to source data we can identify those entries, which have to be updated or deleted within our database.

The mainly autonomous global schema design eases the update process in general because we upload the new release of a data source and are able to separately generate a new target sub-database, which is then connected to the existing data. Otherwise, there would be the necessity to merge the data from the new release with the existing and unchanged data from additional sources to generate a new version of the database.

4.2.3 Usability

In keeping data from the individual sources identifiable within the global schema the users can easily formulate the query on well known concepts. Schema integration over domains with a high degree of semantic heterogeneity like the life science or genome data domain result in an integrated schema which is rather complex (when using schema integration), making the schema difficult to understand. The resulting schema often differs heavily from the original schema, making it hard for the domain expert, used to the original sources, their concepts and terminology, to find their way around.

The multidimensional data integration approach also allows for a very intuitive query model. Queries are designed as refinements or filters over properties of proteins as derived from the different data sources. We expand on this aspect in Section 6.

5 Integration Process and Schema Design

5.1 COLUMBA Schema Design

The resulting schema (Fig. 2) is organized around PDB entries. They are surrounded by information from the additional sources, listed in Section 3, defining certain properties of the structures, contained within the entry. Each of the sources can be viewed as a dimension in the multidimensional model.

We want to stress again the importance of not merging data sources, although this might yield data redundancies within the integrated system. In many cases, meaningful schema integration is not possible because of the different concepts used for representing semantically equivalent information. For example SCOP and CATH, two databases storing classifications about evolutionary and structurally related proteins, use different classification hierarchies and methods resulting in different hierarchies despite many sub-trees of the classification being congruent. These differences would inevitably result in conflicts when both data sources should be integrated to one target table. No matter how these conflicts would be solved, the result would be unrecognizable for at least one of the data sources. Thus, we decided to avoid this problem by storing both hierarchies separately and leave it to the query system and/or the end user to decide, which of the sources is to be used.

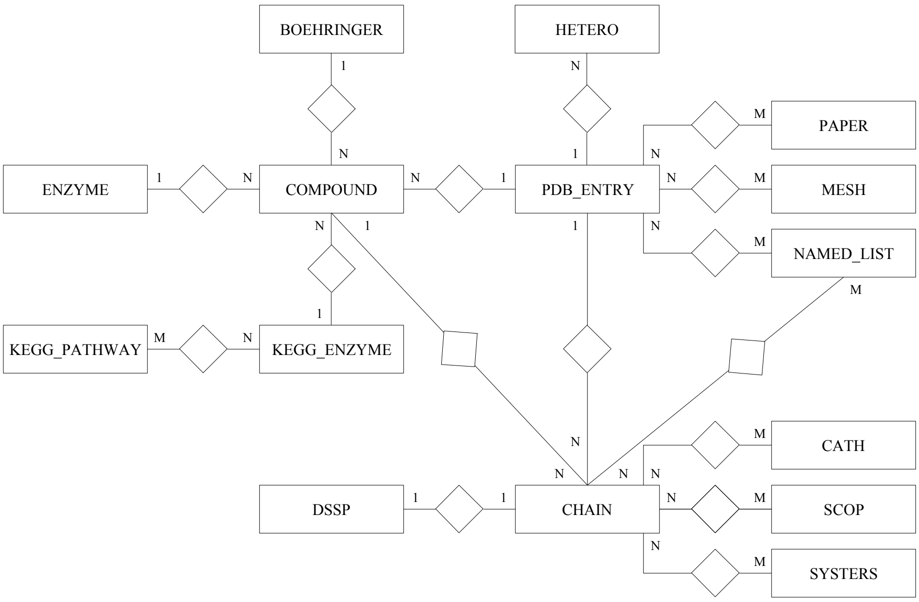


Fig. 2. Entity-Relationship model of COLUMBA. Data in the tables PDB_ENTRY, COMPOUND, CHAIN and HETERO originate from the PDB database, all others from the annotation data sources

5.2 COLUMBA Production Workflow

The production workflow (Fig. 3) of COLUMBA, like the schema, is also centered on PDB entries. Each of theses entries contains information about atom coordinates, macromolecules, their sequence, publications etc. PDB entries are downloaded in the mmCIF format and parsed into a relational schema using the OpenMMS toolkit [22] (Fig. 3.1). This schema is then mapped to a simpler representation (Fig. 3.2).

The annotation pipeline generates connections between PDB entries and objects from the other data sources (Fig. 3.3 / 3.4). Conceptually, each data source is represented by a software module, which implements a fixed interface. After having transformed a new PDB entry into the COLUMBA representation the workflow manager triggers each module by giving them the opportunity to detect and store annotations. Each data source module contains a list of data sources it requires. The workflow manager resolves these dependencies and finds a sequence of handling the modules. To include a new data source, only the tables and the module itself have to be written, and the module name added to a configuration file.

The modules vary according to the nature of the data source. For instance, the DSSP module calls the DSSP program and computes the secondary structure for each chain, whereas the SCOP module only takes the PDB and chain identifiers and matches them with classification files. The entire COLUMBA global schema contains 32 tables and is implemented on PostgreSQL 7.4.

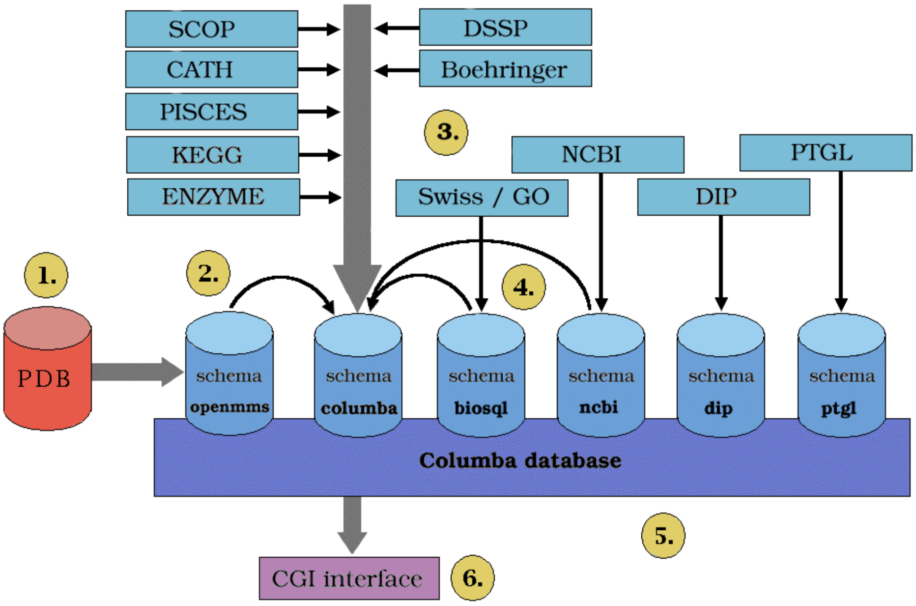


Fig. 3. Production workflow describing data fluxes and content relations: 1.) Load PDB data to OpenMMS database. 2.) Map OpenMMS schema to a simpler model. 3.) Add annotation from multiple sources of data. 4.) Add data to application-specific schemas. 5.) Have everything in one database. 6.) Query the data

5.3 Data Cleansing

The existence of errors and omissions in genome databases is a well known fact. The process of finding and resolving data quality problems in databases is generally called data cleansing [23]. We roughly differentiate between syntax and semantic errors. Syntax errors are mainly domain or format violations in data entries and values as well as misspellings. Syntactic cleansing operations like format and domain transformation, standardization, normalization, and dictionary lookup can be performed within the individual parsers or by separated cleansing modules for each of the sources. Currently, we are only checking for entry and value format violations and correct them by skipping the respective entry or value.

Semantic errors impact the accuracy of data regarding the real-world facts they are to represent. This type of errors is difficult to detect and eliminate. The direction we are currently following for semantic data cleansing utilizes redundant information. Exploiting redundancies for data cleansing is possible in cases where there exist overlapping versions of the same data source with slightly differing content. The differences might be due to manual data cleansing, data transformations, or data enhancement. The PDB database is available in three different versions and formats: (i) the original PDB data available in flat file format, (ii) data in macromolecular Crystallographic Information File format (mmCIF) from the PDB uniformity project at the University of California San Diego (UCSD) aiming at removing inconsistencies in PDB data [24], and (iii) the macromolecular structure relational database (E-MSD), a comprehensive cleansing project at the European Bioinformatics Institute (EBI) to ensure data uniformity and create a single access point for protein and nucleic acid structures and related information, available as Oracle dump files [25].

We currently utilize the parser for PDB flat-files to create an instance of our PDB target schema and the OpenMMS Toolkit, containing software for parsing and loading mmCIF files into a relational database. This toolkit uses a complex schema consisting of approximately 140 tables. We generated a set of schema mapping rules, which transform the data from the OpenMMS schema into a simpler target schema comprising only 6 tables. Thereby, we are able to create two overlapping instances for our PDB target schema. By comparing these instances and highlighting and evaluating the differences, using domain knowledge we are able to identify the reliable parts within both of the instances to select them for integration into a resulting third instance of the PDB target schema.

Identification of corresponding records within the two instances is easily done via the unique PDB identifier forming a matching record pair. We analyze these matching pairs for mismatches in their attribute values. The percentage of mismatches within the attributes varies widely (Table 2). Attributes having close to 100% mismatches often result from different formats or NULL values within one of the instances. Further investigating the mismatch causing values within each of the attributes reveals additional information about the causes for the mismatch.

For instance, comparison and evaluation enabled us to identify 32 records having a deposition year of *1900* in the mmCIF files where the original PDB flat files state the year *2000* for entry deposition. In an other case the structure method for over 2000 of the records resulting from parsing the PDB flat files was *unknown* while the mmCIF files stated *X-ray diffraction* as the structure method used.

We are also investigating mismatch dependencies between attribute pairs (A_i, A_j) where a mismatch in attribute A_i always causes additional mismatches in attribute A_j within the same matching record pair. Identification of these dependencies gives us a hint of the cause of the problem and hence of possible resolution strategies.

We are planning on extending this comparison approach to include E-MSD, giving us three overlapping instances at hand and also on extending the search for patterns of differences within the instances to gain a resulting integrated instance of even higher accuracy for PDB data.

Table 2. Showing for each of the attributes in table PDB_ENTRY the probability $P(A_i)$ for mismatches within matching pairs and the number of different values for each attribute in each of the compared instances PDB-Parsed (resulting from PDB flat files) and MMS (resulting from OpenMMS mapping)

Attribute	$P(A_i)$	PDB-Parsed	MMS
NAME	0.999	2,299	19,736
YEAR_PDB_DEPOSITION	0.004	33	34
DEPOSITION_DATE	0.006	3,755	3,949
RELEASE_DATE	0.572	1,320	1,184
STRUCTURE_METHOD	0.183	135	96
RESOLUTION	0.451	289	356
R_VALUE	0.999	1	851
R_FREE	0.999	1	1,492
REFINEMENT_PROGRAM	1	1	485

6 Web Interface

For accessing COLUMBA a web interface is available at www.columba-db.de. This web interface uses a "query refinement" paradigm to return a subset of the PDB entries. A query is defined by entering restriction conditions on the data source specific annotations (Fig. 4). The user can combine several queries acting as filters to obtain the desired subset of PDB entries. The interface supports interactive and exploratory usage by straightforward adding, deleting, restricting or easing of conditions. For example, the whole set of PDB entries first can be filtered by searching for a name, then a constraint on the resolution of the structures is applied, and finally redundant sequences are excluded. The user is supported by a preview, which constantly shows the number of PDB entries and chains in the result set.

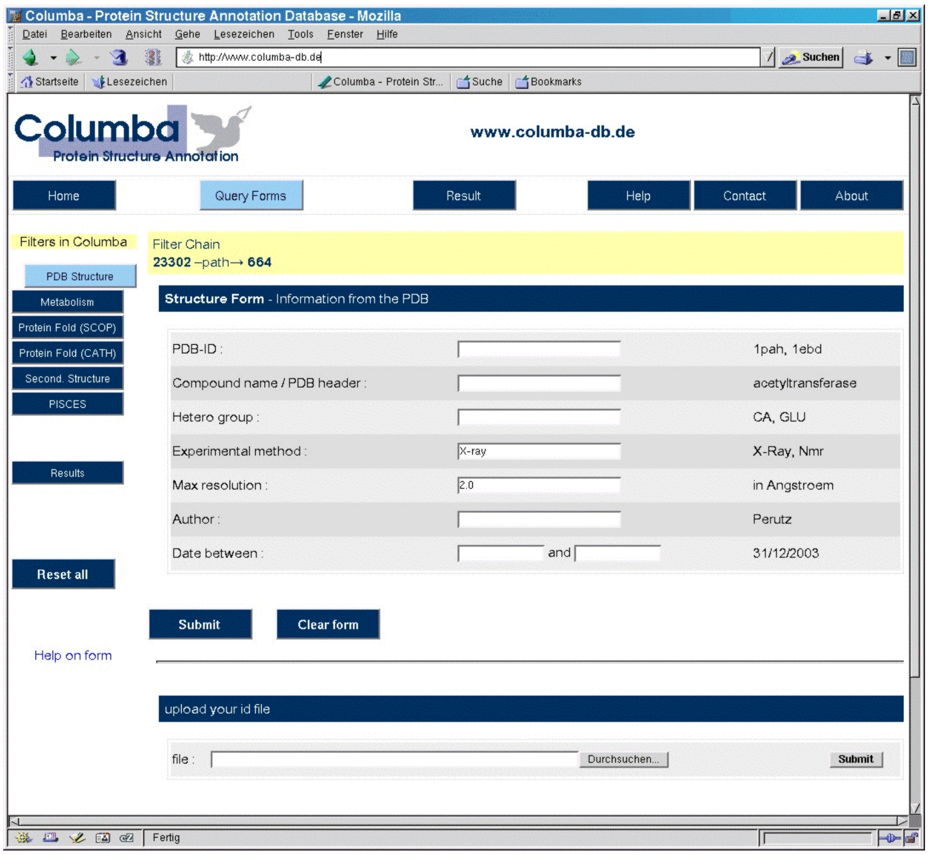
The result set gives basic information to each of the entries returned. This set is available as a formatted HTML table or text file. The user can see the full scope of COLUMBA on a single PDB entry, where all the information from the different data sources is shown.

7 Related Work

There are many efforts providing data on proteins via the Internet, and most of them use relational databases. Like the data sources listed above most of them focus on one special aspect of proteins, thus being an interesting object for integration. Often, they also contain hyperlinks to related databases, following the paradigm of link integration [26].

The PDBSUM [27] service is an excellent example for this approach: It contains a quick summary page for each PDB entry, plus dozens of links. The PDBSUM example is an auxiliary source for gathering information on few structures with known PDB codes, but it is not intended as a query and cross-comparison system.

The Jena Image library [28] provides data on protein structures and some external data sources (NDB, SWISS-PROT) in the form of large ASCII dump files. This approach has an intuitive charm for computer scientists, because parsing does not require much more than to split a string by tabs, but biologists will be deterred. Knowing this, the Jena team has designed a query interface, allowing full text search on all entries.



Filter Chain
23302 –path→ 664

Structure Form - Information from the PDB

PDB-ID :		1pah, 1ebd
Compound name / PDB header :		acetyltransferase
Hetero group :		CA, GLU
Experimental method :	X-ray	X-Ray, Nmr
Max resolution :	2.0	in Angstroem
Author :		Perutz
Date between :	and	31/12/2003

upload your id file

file :

Fig. 4. The COLUMBA Query Interface for specifying restrictions on attributes within the PDB target schema

Finally, there are several integrated databases of protein sequence and genome annotation (e.g. EXPASY [29]). They have been present for years, resulting in mature and reliable web interfaces and tools, which are very frequently used by lab biologists. Although they all contain references to PDB structures, they do neither cover the full spectrum of structure-specific data sources nor the level of detail contained in COLUMBA. The reason for this is obvious: The protein sequence databases are more than ten times larger than the PDB, shifting the focus from the mentioned databases to other fields.

In [30] a data warehouse approach to microarray data is described and also the integration of gene function and gene products is reported. However, no particular integration method is described, which makes a comparison impossible.

The various approaches for schema integration have already been mentioned and advantages and drawbacks discussed. Regarding schema design [31] suggests conceptual models for different types of Life Science data. However, these models are not designed for data integration and no model is proposed for protein structures.

Existing approaches for data cleansing are surveyed in [32]. These approaches currently focus predominantly on syntactical data cleansing while the aspect of semantic data cleansing remains unattended to a great extent. At the moment semantic data cleansing is mostly done using integrity constraint checking for error detection while error correction is often left up to the user. Exploiting overlapping information sources for data cleansing has not been regarded so far.

8 Conclusion

We presented COLUMBA, a database of protein annotations, which currently integrates data from seven data sources including PDB. We described COLUMBA's general architecture, sketched the advantages of the multidimensional approach to data integration implemented in COLUMBA, and discussed first results on data quality estimations we investigate. COLUMBA is freely available on the web for non-commercial and academic usage. Commercial users must obey all license restrictions of each individual data source integrated into COLUMBA. The database is already used intensively in several projects within the Berlin Center for Genome-based Bioinformatics (BCB), such as for the study of DNA-binding proteins within the department for biochemistry of the Charité and for the selection of candidate structures for the evaluation of parallel threading algorithms at the Conrad-Zuse Center.

Acknowledgement

This work is supported by BMBF grant no. 0312705B (Berlin Center for Genome-Based Bioinformatics) and DFG grant number GRK 316 (Berlin-Brandenburg School on Distributed Information Systems). We thank Raphael Bauer, Rene Heek, and Stephan Guenther for implementing many parts of COLUMBA. We truly acknowledge the excellent work of the database maintainer of the different source databases and thank for their consent to integrate their data into COLUMBA.

References

1. Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer, E.F. Jr., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T., Tasumi, M.: The Protein Data Bank: a computer-based archival file for macromolecular structures. J. Mol. Biol., Vol. 112 (1977) 535-542

2. Karp, P. D.: What we do not know about sequence analysis and sequence databases. Bioinformatics 14(9) (1998) 753-754
3. Devos, D. and Valencia, A.: Intrinsic errors in genome annotation. Trends in Genetics 17(8) (2001) 429-431
4. Gilks, W.R., Audit, B., De Angelis, D., Tsoka, S., Ouzounis, C.A.: Modeling the percolation of annotation erros in a database of protein sequences. Bioinformatics, Vol. 18(12) (2002) 1641-1649
5. Kanehisa, M., Goto, S., Kawashima, S., Nakaya, A. The KEGG database at GenomeNet. Nucleic Acid Research, Vol. 30(1) (2002) 42-46
6. Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. Vol. 247 (1995) 536-540
7. Kabsch, W., Sander, C.: Dictionary of protein secondary structure: pattern of recognition of hydrogen-bonded and geometrical features. Biopolymers, Vol. 22(12) (1983) 2577-2637
8. Bairoch, A.: The ENZYME database. Nucleic Acid Research, Vol. 28(1) (2000) 304-305
9. Preissner, R., Goede, R., Froemmel, C.: Dictionary of interfaces in proteins (DIP). Databank of complementary molecular surface patches. J. Mol. Biol., Vol. 280, No. 3 (1998) 535-550
10. Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B., Thornton, J.M.: CATH- A Hierarchic Classification of Protein Domain Structures. Structure, Vol. 5(8) (1997) 1093-1108
11. Boeckmann B., Bairoch A., Apweiler R., Blatter M., Estreicher A., Gasteiger E., Martin M. J., Michoud K., O'Donovan C., Phan I., Pilbout S., Schneider M.: The Swiss-Prot protein knowledgebase and its supplement TrEMBL. Nucleic Acids Research, Vol. 31(1) (2003) 365-370
12. Wheeler, D.L., Church, D.M., Federhen, S., Lash, A.E., Madden, T.L., Pontius, J.U., Schuler, G.D., Schriml, L.M., Sequeira, E., Tatusova, T.A., Wagner, L.: Database resources of the National Center for Biotechnology. Nucleic Acids Res., Vol. 31(1) (2003) 28-33
13. Ashburner M., Ball C.A., Blake J.A., Botstein D., Butler H., Cherry J.M., Davis A.P., Dolinski K., Dwight S.S., Eppig J.T., Harris M.A., Hill D.P., Issel-Tarver L., Kasarskis A., Lewis S., Matese J.C., Richardson J.E., Ringwald M., Rubin G.M., Sherlock G.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nature Genetics 25(1) (2000) 25-29
14. Wang, G., Dunbrack, R.L. Jr.: PISCES: a protein sequence culling server. Bioinformatics, Vol. 19(12) (2003) 1589-1591
15. Krause, A., Stoye, J., Vingron, M.: The SYSTERS protein sequence cluster set. Nucleic Acids Res., Vol. 28(1) (2000) 270-272
16. Michal, G.: Biochemical Pathways. Boehringer Mannheim GmbH (1993)
17. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Wheeler, D.L.: GenBank. Nucleic Acids Res., Vol. 31(1) (2003) 23-37
18. Lakshmanan, L., Sadri, F., Subramanian, I.: On the Logical Foundation of Schema Integration and Evolution in Heterogeneous Database Systems. Intl. Conference on Deductive and Object-Oriented Databases (DOOD) (1993) 81-100
19. Do, H.H., Rahm, E.: COMA - A System for Flexible Combination of Schema Matching Approaches. Conference on Very Large Data Bases(VLDB) (2002) 610-621
20. Rahm E., Bernstein P. A.: A survey of approaches to automatic schema matching. VLDB Journal, Vol. 10 (2001) 334-350
21. Chaudhuri, S., Dayal, U.: An Overview of Data Warehousing and OLAP Technology. SIGMOD Record Vol. 26 (1997) 65-74
22. Greer, D.S., Westbrook, J.D., Bourne, P.E.: An ontology driven architecture for derived representations of macromolecular structure. Bioinformatics, Vol. 18(9) (2002) 1280-1
23. Rahm, E., Do, H.H.: Data Cleaning: Problems and current approaches. IEEE Bulletin of the Technical Committee on Data Engineering, 23(4) (2000)

24.Bhat, T.N., Bourne, P., Feng, Z., Gilliland, G., Jain, S., Ravichandran, V., Scheider, B., Schneider, K., Thanki, N., Weissig, H., Westbrook, J., Berman, H.M.: The PDB data uniformity project, Nucleic Acid Research, Vol. 29(1) (2001) 214-218

25.Boutselakis, H., Dimitropoulos, D., Fillon, J., Golovin, A., Henrick, K., Hussain, A., Ionides, J., John, M., Keller, P.A., Krissinel, E., McNeil, P., Naim, A., Newman, R., Oldfield, T., Pineda, J., Rachedi, A., Copeland, J., Sitnov, A., Sobhany, S., Suarez-Urunea, A., Swaminathan, J., Tagari, M., Tate, J., Tromm, S., Velankar, S., Vranken, W.; E-MSD: the European Bioinformatics Institute Macromolecular Structure Database. Nucleic Acid Research, Vol. 31(1) (2003) 458-462

26.Stein, L.: Creating a bioinformatics nation. Nature, Vol. 417(6885) (2002) 119-120

27.Laskowski, R.A.: PDBsum: summaries and analyses of PDB structures. Nucleic Acids Research, Vol. 29(1) (2001) 221-222

28.Reichert, J., Suhnel, J.: The IMB Jena Image Library of Biological Macromolecules: 2002 update. Nucleic Acids Res., Vol. 30(1) (2002) 253-254

29 Gasteiger E, Gattiker A, Hoogland C, Ivanyi I, Appel RD, Bairoch A.: ExPASy: The proteomics server for in-depth protein knowledge and analysis. Nucleic Acids Research, Vol. 31(13) (2003) 3784-3788

30.Cornell, M., Paton, N. W., Shengli, W., Goble, C. A., Miller, C. J., Kirby, P., Eilbeck, K., Brass, A., Hayes, A. and Oliver, S. G.: GIMS - A Data Warehouse for Storage and Analysis of Genome Sequence and Function Data. 2nd IEEE International Symposium on Bioinformatics and Bioengineering, Bethesda, Maryland (2001)

31.Paton, N. W., Khan, S. A., Hayes, A., Moussouni, F., Brass, A., Eilbeck, K., Goble, C. A., Hubbard, S. J. and Oliver, S. G.: Conceptual Modelling of Genomic Information. Bioinformatics 16(6) (2000) 548-557

32.Müller, H., Freytag, J.C.: Problems, Methods, and Challenges in Comprehensive Data Cleansing. Technical Report HUB-IB-164, Humboldt University Berlin, (2003)