Flexible web-based integration of distributed large-scale human protein interaction maps

Gautam Chaurasia[1,2,*], Yasir Iqbal[1], Christian Hänig[2], Hanspeter Herzel[1], Erich E. Wanker[2] and Matthias E. Futschik[1]

[1]Institute for Theoretical Biology, Charité, Humboldt-University, Berlin, Germany
[2]Max Delbrück Center for Molecular Medicine, Berlin, Germany

Summary

Protein-protein interactions constitute the backbone of many molecular processes. This has motivated the recent construction of several large-scale human protein-protein interaction maps [1-10]. Although these maps clearly offer a wealth of information, their use is challenging: complexity, rapid growth, and fragmentation of interaction data hamper their usability. To overcome these hurdles, we have developed a publicly accessible database termed UniHI (Unified Human Interactome) for integration of human protein-protein interaction data. This database is designed to provide biomedical researchers a common platform for exploring previously disconnected human interaction maps. UniHI offers researchers flexible integrated tools for accessing comprehensive information about the human interactome. Several features included in the UniHI allow users to perform various types of network-oriented and functional analysis. At present, UniHI contains over 160,000 distinct interactions between 17,000 unique proteins from ten major interaction maps derived by both computational and experimental approaches [1-10]. Here we describe the details of the implementation and maintenance of UniHI and discuss the challenges that have to be addressed for a successful integration of interaction data.

1 Introduction

Protein-protein interactions play an essential role in many biological processes. They constitute core processes of the cellular machinery. Comprehensive catalogs of such interactions would facilitate the unrevealing of complex biological mechanisms. Recently, various approaches to systematically map the so-called protein interactomes have emerged. After initial assemblies of interaction maps for model organisms such as *S. cerevisiae*, *D. melanogaster* and *C. elegans* [11-15], the systematic mapping of the human interactome has moved into focus. Both computational and experimental approaches have been used to construct human protein interaction maps [1-10]. However, several comparative analysis of interaction maps [16-18] have shown that the quality of interaction data has to be critically assessed, basically due to the low saturation and high false positive rates.

Nevertheless, currently available human interaction maps offer not only a wealth of information but can also be a great assist to the biomedical research community for systematic analysis of molecular networks. However, utilization of these interaction maps is impeded due to their incompleteness and missing integration, as data on proteins and their interacting partners are distributed across multiple locations [1-3,6-10]. To find comprehensive information on human proteins of interest, scientists may have to perform repeated searches in many databases. This is evidently very time-consuming as various query formats and

* Corresponding author, g.chaurasia@biologie.hu-berlin.de

identifiers have to be used in different databases. Another major limitation in currently available interaction databases is that frequently only interactions for single protein can be queried. However, modern systems biology requires complex network-oriented search for interaction of multiple proteins.

Previous studies have also shown that current human interaction maps are highly divergent [16,17], indicating that these maps contain complementary information and their unification can provide us a deeper understanding of the human interactome. At the same time, information about data quality and validation can help experimentalists to assess critically interactions found in the databases.

Another important issue of integrated databases is their regular updates and extensibility. As we are still very far from the completion in the human interactome, interaction data will grow continuously. Therefore, it is necessary to implement tools that can keep the existing interaction data updated, and enable easy inclusion of newly discovered interactions.

The main challenges discussed above demand: i) comprehensive integration of the currently available human interaction maps; ii) performing network-oriented complex queries; iii) quality assessment of the data; and iv) regular updates and the extensibility of the integrated database. To address these tasks, we constructed a flexible web-based database, termed UniHI. It is intended to provide an integrated platform for finding comprehensive information on human proteins and their potential interaction partners. UniHI houses over 160,000 distinct interactions between 17,000 unique proteins from ten major large-scale human protein interaction maps (see details in Table 1). UniHI provides scientists with a user friendly web-interface available at *http://www.mdc-berlin.de/unihi*.

In this paper, we will emphasize the implementation details and architecture of UniHI. A short overview of UniHI features is also given, while further details on characteristics and statistical analysis of included interaction datasets are described in reference [19]. Finally, we discuss the scope and future direction of UniHI.

2 Architecture of UniHI

The architecture of the UniHI database has been designed to integrate interaction data obtained from different sources, and also to incorporate future human interaction maps, if they become available. The advantage of the UniHI architecture is its modularity and portability by introducing a multi-tier architecture with four separated layers: i) integration; ii) database; iii) persistence; and iv) application. The integration layer is responsible for downloading, parsing, preprocessing and updating of data. The database layer is a relational database which stores and manages the information on proteins and their interaction partners from different sources into one common schema. The persistent layer is used for inserting and retrieving data from the database. The application layer provides a web-interface and a visualization tool with many interactive features for accessing and viewing the interaction data. Figure 1 shows the architecture of the UniHI database. We describe the functionality of each layer in the following sections.

2.1 Integration

Data integration from different data sources imposes major tasks. They include careful assembly of similar and complementary information from heterogeneous data sources and deletion of duplicated data. Such requirements would demand considerable hand coded programming efforts, as different data formats have to be combined into a common schema. In contrast, the object-relational mapping approach (Hibernate) [20] provides mechanisms for

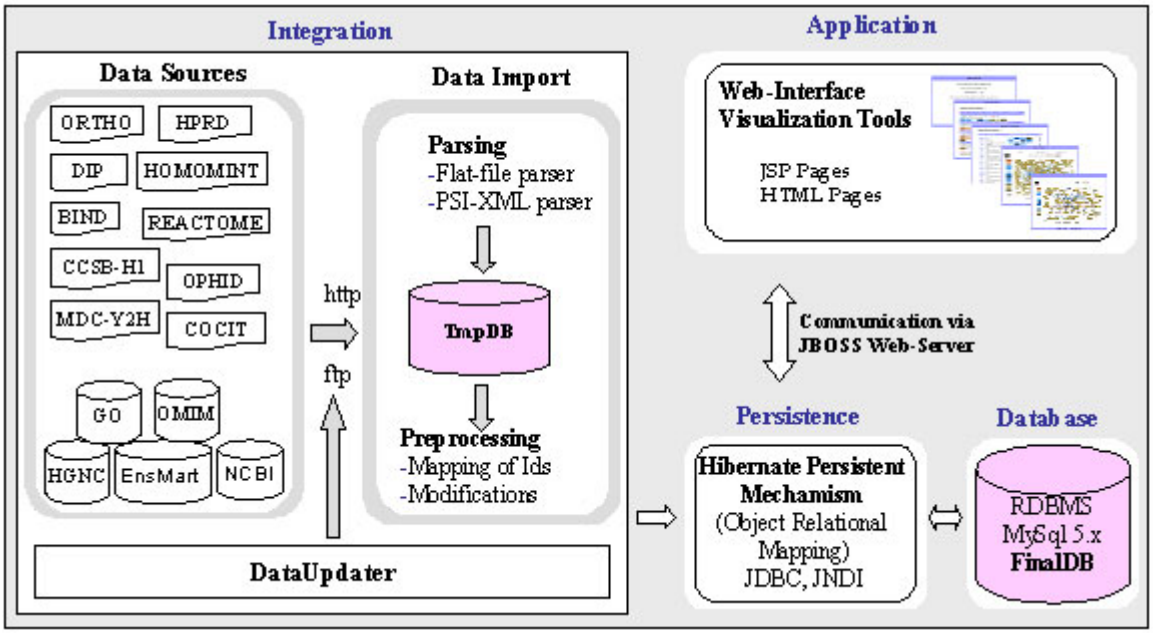

Figure 1: Architecture of the UniHI, consists of four main architectural layers: i) integration, responsible for the data downloading, parsing, preprocessing and updating; ii) database, consists of a relational database which stores and manages the information on proteins and their interaction partners from different sources using one common schema; iii) persistence, used for inserting and retrieving the data from the database via Hibernate persistent mechanism; iv) application, providing a web-interface and a visualization tool for accessing and viewing the interaction data.

automating the whole process, thereby reducing the programming tasks. Details on the different data sources and their integration mechanism are given in the following sections.

2.1.1 Data Sources

Currently, protein interactions in UniHI are derived from ten large-scale human protein-protein inetraction maps [1-10]. These maps were generated either using yeast-two-hybrid (Y2H) assays, literature review or orthology-based approaches. In its first version, UniHI includes five literature-based interaction maps (BIND [1], HPRD [2], COCIT [4], DIP [7] and REACTOME [10]), two Y2H-based interaction maps (MDC-Y2H [5], and CCSB-H1 [6]) and three orthology-based maps (HOMOMINT [3], OPHID [8] and ORTHO [9]). Further details on these maps are given in the table 1. Additional information on proteins (e.g. functional annotations and description of proteins, chromosomal location and potential disease association of corresponding genes) were imported from National Center for Biotechnology Information (NCBI) [21], Online Mendelian Inheritance in Man (OMIM) [22] and Gene Ontology (GO) [23] databases. Lists of different protein identifiers were downloaded from HUGO Gene Nomenclature Committee (HGNC) [24], and EnsMart [25].

2.1.2 Data Downloading and Parsing

Interaction data are downloaded from different distributed sources (see Table 1) via file transfer protocol (ftp) or hypertext transfer protocol (http). The data are generally available in two different formats either as flat-files or XML-files (eXtensible Markup Language) (see Table 1). Most of the interaction databases now release their datasets following the XML-based PSI-MI (Proteomics Standards Initiative - Molecular Interaction) convention [26]. Separate parsers using Java application programming interfaces (APIs) have been

implemented for extracting information from the XML- and flat-files. SAX [27] and DOM [28] parser were used for processing the XML files. The extracted information is imported into a temporary database (TmpDB, figure 1).

2.1.3 Data Preprocessing

As the interaction maps use different identifiers, one of the main challenges in integrating the data is the construction of a unique identifier indexing system. For unification, first, complete lists of proteins for each interaction map were compiled separately. Subsequently, these lists were compared employing information from NCBI, HGNC and EnsMart to map their corresponding identifiers in other interaction datasets. After mapping, identical protein identifiers were merged together in a horizontal manner where each protein is a unique entry in the *Protein* table (see details DB schema). A unique identifier was assigned to each protein entry of this table. These unique identifiers were further used for grouping of the redundant interactions from all interaction datasets. Information on the source of proteins or interactions were merged vertically and inserted into two different tables *ProteinSource* and *InteractionProperties* (for further details see section 2.2 DB schema).

After integration, some modifications on interaction datasets were also performed.

- First, we wanted to distinguish between interactions of binary and complex type. For the binary interaction type proteins interact directly, whereas for complex interaction type proteins belong to the same protein complex but do not necessarily interact directly with others. Most interaction networks include either binary or complex type enabling easy distinction. An exception is HPRD, which provides both binary and complex type of the interaction data. To facilitate differentiation between these two categories, we have split interaction data from HPRD into two sets (HPRD-BIN, HPRD-COMP).
- Secondly, large-scale interaction networks are generally derived by literature-reviews, Y2H-assays or are based on observed interactions between orthologous proteins in other organisms. To indicate users the approach taken those interaction datasets were modified where interaction data were assembled by multiple approaches. For example, OPHID contains orthology-based interactions as well as interactions imported from other databases. We extracted only orthology derived interactions from OPHID as UniHI already includes remaining interactions. Similarly, HPRD contains data from large-scale experiments [5,6] that are separately incorporated in UniHI. Hence, these data were filtered from HPRD interaction map.
- Finally, networks based on multiple approaches were divided according to the method used. CCSB-H1 data were split into Y2H- and literature-based interaction maps (CCSB-Y2H, CCSB-LIT). Table 1 gives a complete overview of the interaction datasets included in the UniHI.

The processed and non-redundant data are inserted into a common relational database using the persistent layer (described in detail in section 2.3).

2.1.4 Data Updates

Another important function of the integration layer is to check for updates. Several of the included interaction data sets are not static, but growing (Table 1). To accommodate this growth of interaction data, regular updates are performed every three months. The updating process is done by a function *DataUpdater*, which initiates Java programs for downloading, parsing and preprocessing of data.

Name	*Ps*	*Is*	*Method*	*Data Format*	*Data State*
MDC-Y2H	1703	3186	Y2H SCREEN	Tab-delimited	Static
CCSB-Y2H	1549	2754	Y2H SCREEN	Tab-delimited	Static
CCSB-LIT	2192	4067	TEXT MINING	Tab-delimited	Static
HPRD-BIN	5908	15508	LITERATURE	PSI-MI-XML	Dynamic
HPRD-COMP	1277	4468	LITERATURE	PSI-MI-XML	Dynamic
DIP	1033	1303	LITERATURE	PSI-MI-XML	Dynamic
BIND	4273	5863	LITERATURE	Tab-delimited	Static
COCIT	3737	6580	TEXT MINING	PSI-MI-XML	Static
REACTOME	679	12639	LITERATURE	Tab-delimited	Dynamic
ORTHO	6225	71466	ORTHOLOGY	XML	Static
HOMOMINT	4127	10174	ORTHOLOGY	PSI-MI-XML	Static
OPHID	4785	24991	ORTHOLOGY	Tab-delimited	Static

Table 1: An overview of all interaction maps included in the UniHI database. The abbreviations '*Ps*' and '*Is*' are the total number of proteins and interactions in the corresponding databases. The column 'Method' lists the approaches used for generating the interaction maps. The column 'Data Format' describes the formats of the available data. The column 'Data State' provides information about state of the included interaction maps.

2.2 Database

Data stored in UniHI are administered by a relational database using an open source MySQL relational database management system (RDBMS) [29]. It consists of nine key tables. The *Protein* table contains a complete list of proteins from all interaction maps. Each protein in this table is stored with its different identifiers (EntrezGene ID, Uniprot ID, Ensembl ID, UniGene ID, OMIM ID) as well as its gene symbol, description, cross-reference database identifiers from HPRD, BIND and DIP, if known. Each protein in this table is a unique entry. The *ProteinAliases* table lists the information of different symbols assigned to the corresponding proteins. The *ProteinSource* table houses information about the occurrence of each protein in different maps. The *GOAnnotation* table stores the information about GO-environment of each protein. The *Interaction* table contains information on interactions. Each interaction in this table is a unique entry. The *InteractionProperties* table gives additional information about interactions such as source of interaction and quality score. For example, interactions of the MDC-Y2H map were categorized as low, medium and high confidence by the authors [6]. The *InteractionScore* table includes information about co-expression and co-annotation of interacting proteins. The *ExperimentDetail* and *DetectionMethod* tables store the information about the Pubmed IDs and the methods used to detect the interactions. The schema for relational database is presented in figure 2.

2.3 Persistence

The persistence layer is the core of the whole system and works as middleware for inserting and querying data. All objects implemented within this layer are mapped to tables in the SQL-based relational database. The role of all objects and their event classes are described in Hibernate mapping properties files. These mapping files are used for the communication with

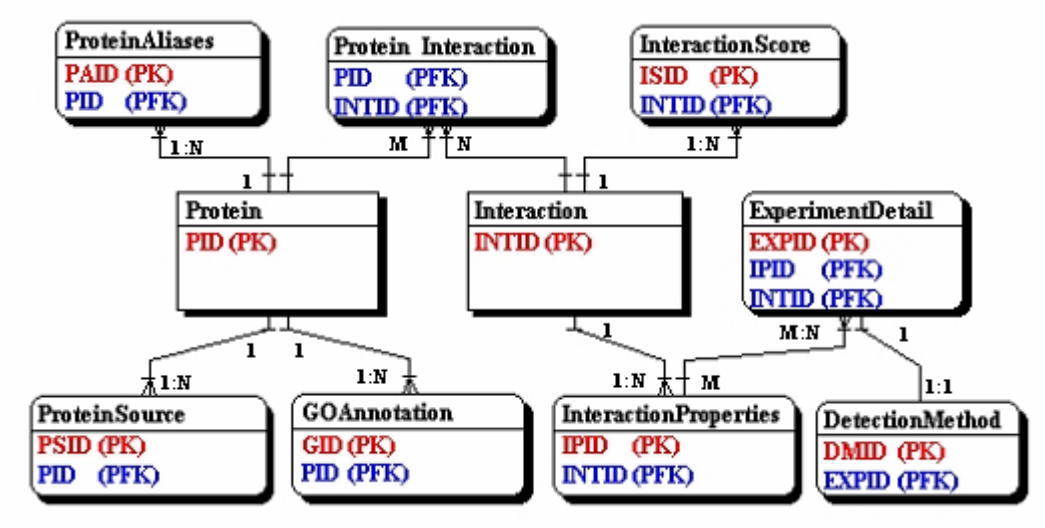

Figure 2: An entity-relationship diagram of UniHI, showing key tables (rectangles) and relations (lines). PK (red) and PFK (blue) denote primary and foreign keys.

the database via Hibernate persistent mechanisms [20]. One of the main advantages of Hibernate is that it automatically generates the SQL calls for querying the data. Complex SQL queries can be simply performed by a single line command. This architecture reduces development time and also optimises the performance.

2.4 Application

The application layer is implemented using web-services of the J2EE architecture. The main purpose of this layer includes communication with clients via a JBOSS web-server [30] and retrieval of the data from the database using hibernate persistent mechanism. Data retrieval is carried out using the Hibernate Query Language (HQL), which is implemented in a set of Java APIs. Further, this layer consists of a web interface and a visualization tool. Functions included in this interface enable users to perform not only simple searches for interactions of single protein but also complex network-oriented queries for multiple proteins. It provides additionally several features for refined search and selective use of interaction maps. Validation schemes provided with each interaction map are also included to assess the quality of each interaction. The various features of interface and visualization tool are detailed in the next sections.

2.4.1 Web-Interface

UniHI web-interface provides the user with two different search options: i) Single protein search; and ii) multiple protein search. In a single protein search, users provide a single protein to query for its direct interaction partners. In a network-oriented multiple protein search, users can supply a list of proteins. Interactions can be queried using following protein identifiers: EntrezGene ID, Uniprot ID, Ensembl ID, Unigene ID, NCBI Geneinfo ID, OMIM ID and Gene Symbol. Figure 3 shows an example, illustrating two search schemes in the UniHI database: i) sequence of the main search; and ii) sequence of optional search. This division allows users to retrieve further information if demanded. At the same time, the amount of information displayed can be limited to the level required by the user. Using the optional search tools, various statistical analyses can be accessed regarding network structure, co-expression and functional annotations. Additionally, the interaction maps included were rigorously examined regarding network structure and supported by independent data [19].

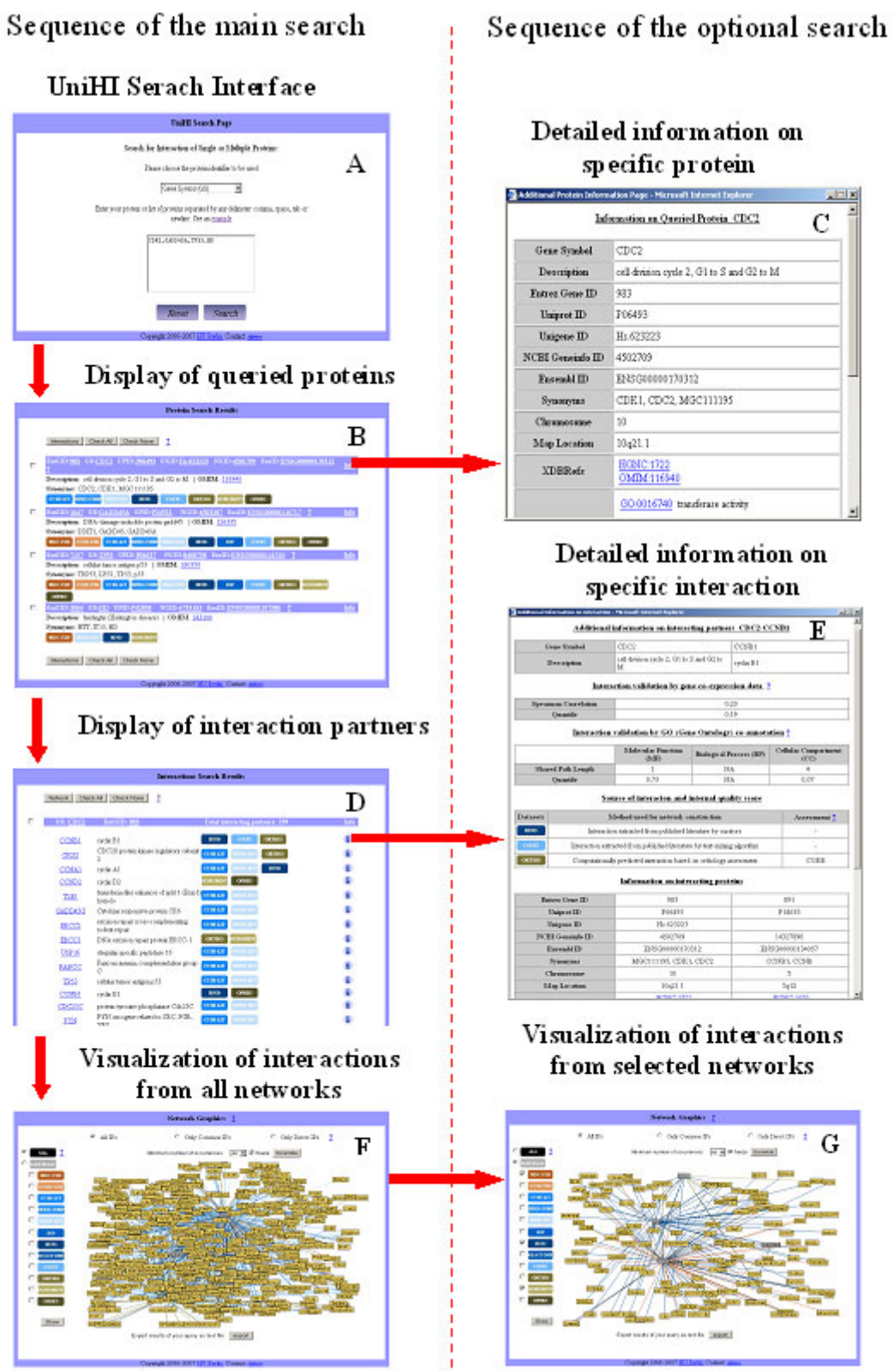

Figure 3: The workflow of the UniHI database can be divided into two layers: i) sequence of the main search (windows A, B, D and F); and ii) sequence of the optional search (windows C, E and G). This is illustrated here using the following exemplary set of proteins: CDK1, GaDD45A, TP53, HD. Starting with the window A, a list of protein identifiers is given in UniHI search tool. Results from this search are displayed in window B, where all queried proteins are listed with their corresponding information e.g. synonyms, functional description and the distribution in different interaction data sets. Further information on proteins can be found by following the link "Info" opening window C. Now user can view the interaction partners of all proteins or only

of selected proteins from check box list. Window D list the interacting partners with additional information on the distribution of each interaction in different data sets. The total number of interacting partners indicates the connectivity of the particular protein. Each interaction in the list is further linked to the databases where it originates from. The quality of each interaction can be assessed by several validation schemes provided with each interaction (window E). Window F is the graphical presentation of the interaction data found in all interaction maps. Query proteins are displayed in gray color, whereas interaction partners are shown in yellow color. However, user can also perform network specific searches by selecting the option "Individuals" followed by network selection from a checkbox list (window G).

2.4.2 Visualization Tool

Graphical representation of the interaction data facilitates the understanding of the network structure. Thus, we have implemented a visualization tool including various features. It is a further extension of a pre-existing Java applet for graphical presentation of interaction networks [31]. Features included in this tool are: i) display of all interaction; ii) display of interaction from selective networks; iii) display of common interacting partners; iv) display of only direct interacting partners. An example of the flexibility of network visualization is also given in figure 3. Further description of UniHI features and their applications can be found in reference [19].

3 Implementation

As the amount of data on protein interactions is growing rapidly, there is ongoing demand for integrated platforms with a high degree of flexibility. Such platforms should not be only easily accessible but also be consistently updated. Data should be accurately integrated from different sources and queries should be processed in minimal time. The structure of the platform should be extensible to new data without changing its data structure. Thus, a careful design and implementation of the system and the selection of computational approaches to assemble heterogeneous data sources are crucial. Traditional computational approaches like object-oriented software and relational databases can be cumbersome and time-consuming. Typically, persisting data objects from SQL tables with a JDBC (Java Database Connectivity API) connection and prepared SQL statements may be easy for simple objects, but is very complicated for objects with many properties such as proteins and their interaction partners, since they have to be mapped to different domains of similar and complementary information. Thus, we decided to implement UniHI with an object/relational mapping (ORM) methodology [20]. ORM tools provide an easy-to-use framework for mapping an object-oriented domain model to a traditional relational database. This technique helps us to reduce the implementation costs of complex SQL queries. ORM takes plain Java objects used in the application and process them using a persistent mechanism which automatically generates all the SQL command needed to store and retrieve the object. Applications built with an ORM tool are cheaper to design, better performing, highly portable and resilient in the face of changes to internal objects or underlying relational models.

The implementation of UniHI uses the ORM tool Hibernate (*http://www.hibernate.org*) [20], an J2EE (Java 2 Platform Enterprise Edition) architecture (*http://www.java.sun.com*) and the relational database management system MySql (*http://www.mysql.org*) [29]. All employed tools are free, open source software and distributed under the General Public License (GNU). Other tools used in the implementation were SAX [27] and DOM [28] parsers. SAX is a Simple API for reading and processing XML data. DOM (Document Object Model) is an abstract data structure that represents XML documents as trees of nodes.

All services run on a Linux system and communicate with clients via a JBOSS web-server (an open source web-application server implemented in Java). This system is completely web-

based and platform independent, and can be accessed using any operating system with an included web-browser.

4 Conclusions and Future Directions

Systems biology has witnessed a rapidly growing number of human interaction maps in recent years. Although these maps offer valuable information, their application is still limited due to the lack of integration. There is clear necessity to have integrated tools, which provide direct access to distributed interaction data at one common platform. Therefore, we developed a flexible web-based database that integrates the human interaction data from ten major sources. The architecture and implementation of UniHI aims to overcome the major challenges in the use of current interaction maps, i.e. rapid growth, fragmentation and complexity of data. UniHI provides researchers with a flexible integrated tool for finding and using comprehensive information about the human interactome. Several features included in UniHI enable researchers to perform network-oriented and global analysis of the human interactome. Several validation schemes have been provided to assess the quality of each interaction. The multi-tier architecture of UniHI facilitates its further extension for future interaction maps without modification of its existing structure. Interaction maps included in UniHI will be regularly updated. Notably, UniHI is not aimed to replace the separate interaction maps, but to provide direct access to an integrated human interactome. In the near future, we also like to integrate the information from protein family databases such as Pfam [32] and gene expression databases like Gene Expression Omnibus (GEO) [33] and ArrayExpress [34]. We hope that this unified database can provide a convenient platform to support scientists undertaking large-scale systems biology.

5 Acknowledgement

The construction of UniHI was supported by SFB 618 grant of the *Deutsche Forschungsgemeinschaft (DFG)*.

6 References

[1] Alfarano C. *et al.* (2005), The Biomolecular Interaction Network Database and related tools, update, Nucleic Acids Research, 33:D418–D424, Database issue.

[2] Gopa R. *et al.* (2006), Human protein reference database, update, Nucleic Acids Research, 2006, Vol. 34:D411–414, Database issue.

[3] Persico M. *et al.* (2005), HomoMINT: an inferred human network based on orthology mapping of protein interactions discovered in model organisms, BMC Bioinformatics, 6 Suppl 4, S21.

[4] Ramani A.K. *et al.* (2005), Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome, Genome Biology, 6, R40.

[5] Rual J.F. *et al.* (2005), Towards a proteome-scale map of the human protein-protein interaction network, Nature, 437, 1173-1178.

[6] Stelzl U. *et al.* (2005), A Human Protein-Protein Interaction Network: A Resource for Annotating the Proteome, Cell, 122, 957-968.

[7] Salwinski L. *et al.* (2004) The Database of Interacting Proteins, update. Nucleic Acid Research, 32:D449-51, Database issue.

[8] Brown K.R. and Jurisica I. (2005), Online Predicted Human Interaction Database, Bioinformatics, 21, 2076-2082.

[9] Lehner B. and Fraser A.G. (2004), A first-draft human protein-interaction map Genome Biology, 5, R63.

[10] Joshi-Tope G. *et al.* (2005), Reactome: a knowledgebase of biological pathways, Nucleic Acids Research, 33:D428-432, Database issue.

[11] Gavin A.C. *et al.* (2002), Functional organization of the yeast proteome by systematic analysis of protein complexes, Nature, 415, 141-147.

[12] Giot L. *et al.* (2003), A Protein Interaction Map of *Drosophila melanogaster,* Science, 302, 1727-1736.

[13] Ito T. *et al.* (2001), A comprehensive two-hybrid analysis to explore the yeast protein interactome, Proc Natl Acad Sci USA, 98, 4569-4574.

[14] Li S. *et al.* (2004), A Map of the Interactome Network of the Metazoan *C. elegans*, Science, 303, 540-543.

[15] Uetz P. *et al.* (2000), A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae,* Nature, 403, 623-627.

[16] Chaurasia G. *et al.* (2006), Systematic Functional Assessment of Human Protein-Protein Interaction Maps, Genome Informatics, 17(1), 36-45.

[17] Futschik M.E. *et al.* (2006), Comparison of Human Protein-Protein Interaction Maps, Lecture Notes in Informatics, P 83, 21-32.

[18] von Mering C. *et al.* (2002), Comparative assessment of large-scale data sets of protein−protein interactions, Nature, 417, 399-403.

[19] Chaurasia G., *et al.* (2007) UniHI: An entry gate to human protein interactome, Nucleic Acid Research 35 Database issue:D590-4.

[20] Documentation on Hibernate: http://www.hibernate.org/5.html

[21] National Center for Biotechnology Information: http://www.ncbi.nlm.nih.gov/.

[22] Hamosh A., *et al.* (2005), Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders, Nucleic Acids Res. 1; 33: D514–D517, Database Issue.

[23] Midori A., et *al.* (2006), The Gene Ontology (GO) project in 2006, Nucleic Acids Res. 34:D322-D326, Database issue.

[24] Eyre T.A. *et al.* (2006), The HUGO Gene Nomenclature Database, 2006 updates, Nucleic Acids Research, 34:D319-D321, Database issue.

[25] Kasprzyk A. *et al.* (2004), EnsMart: a generic system for fast and flexible access to biological data. Genome Research 14(1):160-9

[26] Hermjakob H. *et al.* (2004), The HUPO PSI's molecular interaction format--a community standard for the representation of protein interaction data. Nature Biotechnology 22(2): 177-83.

[27] SAX: A simple API for XML: http://www.saxproject.org/

[28] DOM: Document Object Model: http://www.w3schools.com/dom/dom_parser.asp

[29] Documentation on MySql: http://mysql.org/doc/#manual

[30] Documentation on JBOSS Web-server: http://labs.jboss.com/portal/

[31] Mrowka, R. (2001), A Java applet for visualizing protein–protein interactions Bioinformatics, 17, 669-671.

[32] Robert D, *et al.*, (2006), Pfam: clans, web tools and services, Nucleic Acid Research, 34 Database Issue D247-D251

[33] Ron Edgar, *et al.,* (2002) Gene Expression Omnibus: NCBI gene expression and hybridization array data repository, Nucleic Acid Research, Vol. 30: 207-210.

[34] Alvis Brazma, *et al.*, (2003) ArrayExpress—a public repository for microarray gene expression data at the EBI, Nucleic Acids Research 31(1): 68–71.