PiPa: Custom Integration of Protein Interactions and Pathways

Sebastian Arzt, Johannes Starlinger, Oliver Arnold,
Stefan Kröger, Samira Jaeger, Ulf Leser

Knowledge Management in Bioinformatics
Humboldt-Universität zu Berlin
Unter der Linden 6, 10099 Berlin, Germany
{arzt, starling, arnold, kroeger, sjaeger, leser}@informatik.hu-berlin.de

Abstract: Information about proteins and their relationships to each other are a common source of input for many areas of Systems Biology, such as protein function prediction, relevance-ranking of disease genes and simulation of biological networks. While there are numerous databases that focus on collecting such data from, for instance, literature curation, expert knowledge, or experimental studies, their individual coverage is often low, making the building of an integrated protein-protein interaction database a pressing need. Accordingly, a number of such systems have emerged. But in most cases their content is only accessible over the web on a per-protein basis, which renders them useless for automatic analysis of sets of proteins. Even if the databases are available for download, often certain data sources are missing (e.g. because redistribution is forbidden by license), and update intervals are sporadic.

We present PiPa, a system for the integration of protein-protein interactions (PPI) and pathway data. PiPa is a stand-alone tool for loading and updating a large number of common PPI and pathway databases into a homogeneously structured relational database. PiPa features a graphical administration tool for monitoring its state, triggering updates, and for computing statistics on the content. Due to its modular architecture, addition of new data sources is easy. The software is freely available from the authors.

Keywords: Data Integration, Protein Database, Bioinformatics

1 Introduction

Systems Biology is about studying the interplay of entities in complex organisms. Accordingly, the physical relationships that may hold between biological objects are a central element. Such relationships may, for instance, be the formation of protein complexes, the regulation of genes, the binding of small molecules to proteins, cleavage or (de-)phosphorylation of proteins, etc. Biomolecular relationships, especially, are studied with respect to the way they form complex systems, usually called networks. Important classes of biological networks are, among others, metabolic pathways, signaling cascades protein interaction networks and gene regulatory networks.

For each of these types of data, a large number of specialized databases exist. This paper

is about integrating information from various such databases for two particular important types of information: Protein-protein interactions (PPI) and biological pathways.

Data about PPI is available from dozens of databases [LS09]. Examples are IntAct, DIP, MINT, HPRD or BioGrid. Data enters those systems in various ways. Important sources are results from high-throughput experiments such as Microarray, Deep Sequencing or Yeast-2-Hybrid screens. Another important source of information is the scientific literature, which, to this aim, is screened by database curators [PNA+03]. However, policies for selecting data differ, for instance, in the species, the estimated quality, the type of the experimental evidence, or just by chance as curators select papers for being read using non-disclosed strategies. Databases vary greatly in terms of amount of contained interaction, level of available details on each single PPI, access methods etc. A particular issue is the usage of different licenses, which determine the type of work for which the content may be used and whether or not data from a database may be redistributed.

The situation is similar for biological networks [Sch04]. Again, there exist many different databases with different scope and different coverage. The situation might be even slightly worse than for PPI as a biological pathway is a fairly complex object which can be modeled in various ways; accordingly, there is also considerable heterogeneity in terms of schema (or model). Frequently used resources are KEGG, Reactome, and TransPath.

This distribution and heterogeneity is a problem for many types of analysis in Systems Biology. Very often, analysis methods perform the better, the more comprehensive their available input data set is . Pathways and PPI are also very often studied together; for instance, PPIs are used to augment the notoriously incomplete pathway datasets with further proteins to allow for a more comprehensive study of the functional implications of expression experiments [EMXY00]. In our own research, we use pathways and PPI, for instance, to assess the quality of gene clusters [GWPL08] or to enhance the quality of function prediction methods [JGLRS08]. Also, integrated data sets are not only more comprehensive, but the redundancy in their content may be used as a quality indicator for scoring data items [CMR+09]. Accordingly, several projects have emerged that target the integration of PPI and pathway databases. For instance, HPD is an integrated pathway database focusing on human and using a data warehouse approach [CWZ+09]. HAPPI [CMH09], UniHi [CMR+09], PIANA [AJO06] and PID [SAK+09] are integrated PPI databases, all but PIANA focusing on human only. ConsensusPathDB is an integrated database uniting data from PPI databases and from pathway databases [KWLH09]. However, these resources are of limited use to researchers that want to build their own analysis algorithms using a comprehensive dataset. First, many integrated systems only support browsing and do not offer downloads of the database content. Second, many systems employ complex and task-specific data selection procedures, leading to an incomplete coverage of the integrated sources. Third, even if the database content is available for download, it often excludes certain sources due to licensing issues. As an example, the downloadable files from ConsensusPathDB do not contain data from Transfac and KEGG, as redistribution of this data is prohibited by license. Finally, update intervals of these systems often are irregular and, naturally, not adjustable by users.

In this paper, we present PiPa, a system for building an integrated PPI and pathway database. PiPa is meant to be used as a data infrastructure providing up-to-date and crucial

information on two important types of biological entities. It is not a complete information system in itself, as it, for instance, does neither offer any analytical functions nor a graphical query or visualization interface. PiPa comes with a stand-alone administration tool to control data import and to show the status of the database. Furthermore, in contrast to many other integrated databases in this field, PiPa does not perform any semantic integration itself; instead, data from the sources is integrated as such into the system (for instance, no duplicate detection is performed), leaving the decision onto which form of aggregation or quality filtering to perform to the user.

The probably most similar project to PiPa is ATLAS [SHX+05], a system for building a local data warehouse that integrates various types of biological information. However, ATLAS has a much broader scope than PiPa, which in turn leads to much less depth of information. Furthermore, ATLAS is not available any more. Note that PiPa, in contrast to pure synchronization tools such as BioMAJ [FBA+08], not only keeps local copies of remote flat-file representations of biological databases, but also parses these files and loads them into a uniform relational schema.

2 PiPa Data Model

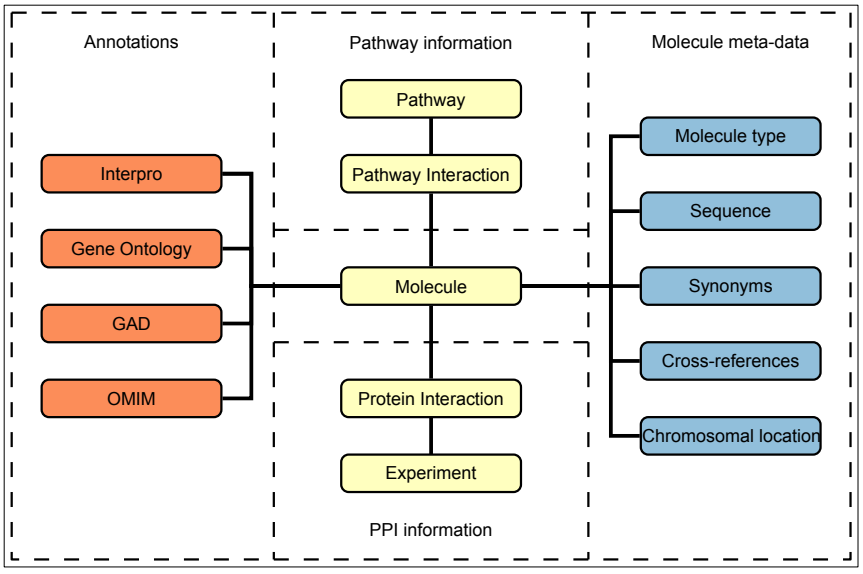

Figure 1: Conceptual data model of PiPa.

The PiPa data model (see Fig. 1) tries to find a balance between simplicity and expressivity. We refrained from adopting complex models (such as, e.g., [BQG+06]), especially in the area of pathways, and only take over the most essential information from the different data providers. With essential, we mean important for performing functional studies on the

data. Accordingly, projects working, for instance, in pathway reconstruction might find the level of detail it offers insufficient.

The essential information we model are proteins, their interactions, and their participation in pathways. Proteins (stored in the molecule table) are annotated with important attributes such as their sequence or chromosomal location and are linked to further information describing their function, domains in their primary sequence, and associations to diseases. Interactions are stored as links between proteins and are annotated with the type of experiment that was used to confirm the interaction and with references to PubMed. Finally, pathways are modeled as a set of interactions, each connecting two chemical entities, which can either be proteins, compounds, or smaller molecules. Furthermore, all interactions and pathways are annotated with the database they were imported from. We also keep external IDs allowing to link back to the original information.

3 Filling and monitoring PiPa

PiPa is a tool providing a graphical user interface for building and updating a local database. Its main function is the import of external data sources into a unified schema. Each import is performed by a source-specific software module implementing a common interface. New modules can easily be plugged into the system to extend PiPa to further sources (see Section 4). Once the relational schema of PiPa is installed in a local database (we currently support MySQL), all sources for which a plug-in is provided can be imported.

PiPa comes with a simple administration tool (see Fig. 2), that provides functions for monitoring the status of the database, for triggering integration / update of data, and for computing basic statistics of database contents.

Source updates always are performed in the same manner. First, source files are downloaded, uncompressed (if necessary), and parsed. The essential information is extracted and inserted into the database. If those data refer to entities that are not presented yet (in particular, new PPIs or pathways may refer to proteins not yet stored in the system), a cascading update is triggered in which the new entity is created and its annotations are downloaded dynamically (also see Section 4). Updates may be interrupted at any time, in which case the entire process is rolled back to ensure data consistency. Furthermore, the date and version of the last update are kept and are displayed in the tool, thus providing a quick overview over the timeliness of the data. Table 1 gives an overview of the data sources currently available in PiPa.

The statistics tab provides aggregated information about the current extent of the database (see Fig. 3). This includes the total number of loaded entities (e.g. molecules, molecules of a certain type, PPIs, pathways etc.) and information on a per-source basis. For PPIs there are per-species-subsections providing information grouped by species. PiPa currently loads data for yeast, nematode, fruit fly, human, mouse and rat. Altogether, PiPa (after a full import of all sources) provides information on app. 25.000 pathways and 12.000.000 molecular relationships, of which app. 2.800.000 are PPIs (the others stem from other compounds involved in a pathway).

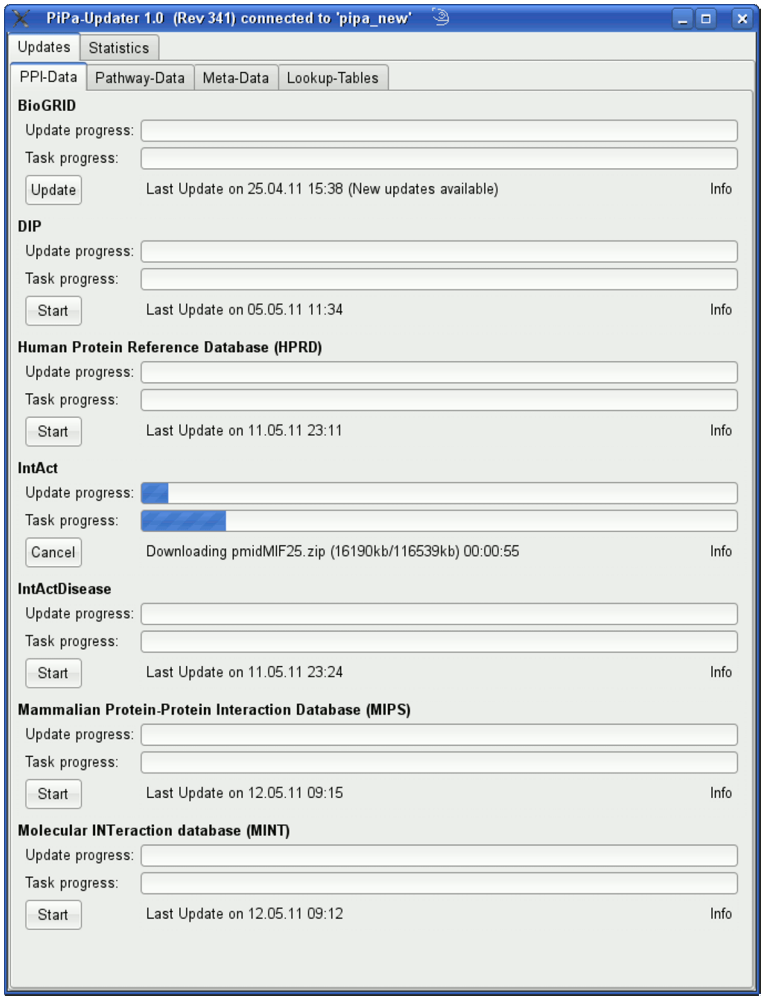

Figure 2: Screenshot of PiPa's administration tool, showing the page for triggering updates of data sources.

4 Implementation

PiPa is written in Java. It uses a plugin-based application design, i.e., each data source is represented by a wrapper module which can easily be plugged in to PiPa's core architecture. Java's object orientation greatly simplifies plug-in development by facilitating the creation of abstract base classes. These base classes implement a rich set of core functions for common tasks, such as downloading and extracting files, keeping track of update times, allocating a database connection, or persisting configuration information in a designated table inside the database. Plug-ins subclass one of these base classes and provide configuration parameters by instantiating variables. These parameters include the location

Pathways	PPI	Protein meta-data
BioCyc	BioGrid	Gad
Inho	DIP	GO
KEGG	HPRD	Interpro
Pathway Commons	IntAct	KEGG
PID	MINT	OMIM
Reactome	MIPS	Reactome
Spike		UniProt

Table 1: List of data sources currently integrated in PiPa.

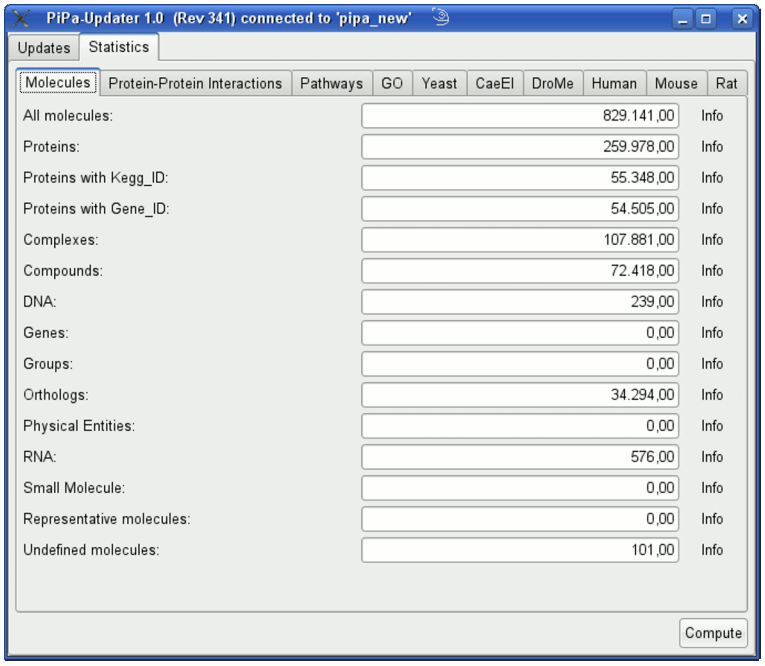

Figure 3: Screenshot of aggregated statistics on the database content. Data can be drilled-down to species-specific information using the respective tabs.

of files to be downloaded and informational messages delivered to the user of the tool. Furthermore, they implement the specific treatment of the particular source.

Additional source-specific code and implementation effort varies greatly between sources. Especially the complexity of parsing is different depending on the type of data stored in a source. For instance, all pathway sources PiPa currently imports offer a BioPax [BBC+05] export; that enables the usage of one and the same parser for all pathway sources. The same applies to PPIs, which are available as PSI-MI XML [HMPB+04] files from all currently integrated PPI sources. In contrast, sources of protein meta-data tend to use proprietary data formats which require their plug-ins to implement parsing capabilities of their own.

A special configuration file determines whether plugins are enabled or disabled (see Fig. 3). Only listed plug-ins listed are visible through the administration interface. The configuration file also manages the concrete assembly of the statistics tab, that can be changed quickly by defining or altering sections, subsections, descriptions, and tooltips for so-called "statistical values". A statistical value can be the result of any singular SELECT-query returning a single value, typically using aggregate functions. Queries are also part of the configuration information and may have parameters. As the computation of the statistical values must be triggered manually from the administration tool, queries can be arbitrarily complex without compromising start up times of the user interface.

As mentioned before, PiPa does not perform extensive semantic integration but keeps information attached to their origin. However, to allow for integrated analysis, we perform ID resolution by normalizing all proteins to their UniProt ID (see Fig. 4). To this end, each protein is handed to the ProteinLoader. If the source does not provide a UniProt ID, the ProteinResolver is asked to map the protein to a UniProt ID, using other IDs given by the source. Currently supported IDs are those from DIP, Ensembl, EntrezGene, EntrezProtein, FlyBase, GeneBank, HGNC, IPI, MGI, OMIM, PDB, Protein gi, RefSeq, SGD, UniParc and WormBase. Once a UniProt ID has been attached to the protein, UniProt is queried for additional data using UniProtJAPI [PWK+08]. If this request is unsuccessful, the UniProt ID itself is subjected to further inspection. First, local lookup tables are used to determine whether the given UniProt ID has been demerged. Demerging of IDs is done by UniProt when a single protein entry is found to refer to more than one real protein. If such a demerger is detected by the ProteinLoader, the original source protein is replaced by the proteins referenced by the new UniProt IDs. If this local lookup is unsuccessful, the UniProt website is queried directly via HTTP to check whether the given UniProt ID is a secondary ID referencing another, primary ID for the same protein (UniProtJAPI does not allow searching with secondary IDs). If a new ID can be assigned in this way, the protein's data is retrieved, again using the UniProtJAPI. Otherwise, if no UniProt ID can be assigned to the protein it is left out and disregarded in the integration process.

In contrast to semantic integration, PiPa's data model (Fig. 1), does provide schematic integration of PPI and pathway sources. In terms of extensibility, plug-ins for such sources thus do not need to provide any details on the target schema. On the other hand, sources for meta-data are typically required to supply their partial schema in the form of SQL statements which extend the data model to include their specific information.

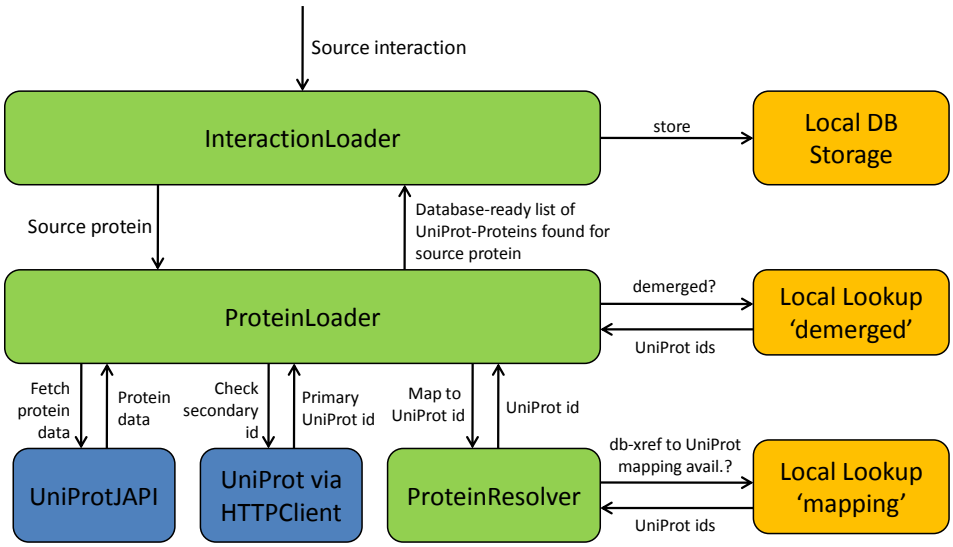

Figure 4: ID normalization for proteins in PiPa.

5 Application of PiPa

PPI represent one of the most important types of biomolecular relationships as virtually all cellular mechanisms rely on the physical binding of two or more proteins for accomplishing a particular task. Protein interactions are essential for controlling cellular processes, such as signal transduction, gene regulation, cell cycle control and metabolism. Numerous experimental methods have been developed for identifying PPIs [PF95]. Large-scale experiments in different model organisms as well as human contributed to an increasing number of comprehensive interaction data sets. Such data sets may be used to identify functional modules within protein networks [DKR+08], to find protein complexes [SM03], or to determine evolutionary conserved processes [SSK+05, JL07]. However, high quality PPI data sets serve different purpose, their are used as input resource for various methods as well as support evaluation strategies for text mining approaches.

5.1 Impact on function prediction

Protein interaction data provide also an important source for functional information. Physical interaction depict, in contrast to sequence, a complementary type of function describing the role of a protein within cells rather than its specific biochemical activity. Furthermore, physically interacting proteins tend to be involved in the same cellular processes, thus interactions represent direct and robust manifestations of functional relationships.

Therefore, protein interaction data are ideally suited to form the basis for function prediction methods.

A wide range of methods has been developed for studying protein interactions in order to predict protein function [SUS07]. Most of them rely on the concept of guilt-by-association, where a protein is annotated based on the function of its interaction partners as illustrated in Figure 5.

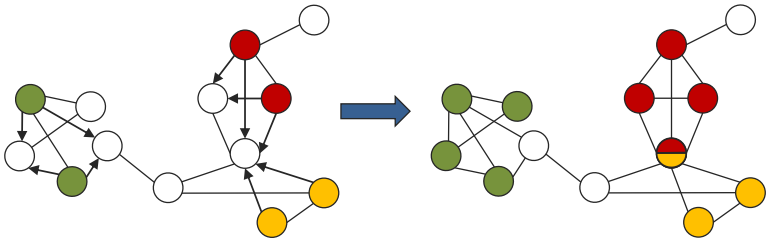

Figure 5: Concept of guilt-by-association for protein function prediction (adapted from [SUS07]). Proteins with known functions are indicated by different colors while proteins without function remain uncolored. Protein function is inferred by transferring functional annotation from directly interacting proteins (indicated by arrows).

The applicability and benefit of protein interaction data for protein function prediction has been established by many approaches [VFMV03, CSW07, JSL10]. Yet, most methods suffer from poor data quality and coverage resulting from systematic and methodological limitations of the respective interaction detection methods. Performance of prediction methods relies largely on the completeness and accuracy of the underlying data. Missing protein interactions, for instance, hinder the prediction process as large fractions of annotated proteins without available interaction data are neglected. High levels of false positive interactions, on the other hand, induce functional associations without biological relevance which reduces the level of accuracy.

Using consistently integrated protein interaction data as provided by PiPa greatly helps to diminish such limitations and thus improves the prediction performance considerably. For instance, more complete interaction data increase the coverage of prediction methods while higher quality increases prediction precision. A rather high coverage of interaction data is achieved by combining various data sets from different data sources. High data quality is obtained by normalizing interacting proteins accurately to avoid redundant information in the data. Further, protein interaction networks can be assembled according to additional evidence associated with an interaction by filtering, for instance, for particular detection methods associated with high levels of false positives or for the experimental role of a protein in an experiment. In an actual application scenario the number of protein functions predicted within a sparse human interaction network with about 13,494 proteins and 43,637 interactions increases from 12,317 to 27,099 when considering a much denser network with 14,218 proteins and 81,868 interactions from PiPa. At the same time, precision increases from 76% in the former network [JSL10] to 83% in the latter one (unpublished).

5.2 Impact on disease gene identification

Protein interactions do not only indicate similar function but often imply common disease phenotypes as gene products associated with a particular disease interact preferentially with proteins known to be involved in the same disease [IS08]. For identifying novel disease gene associations several approaches exploit protein interaction data by growing a network around disease-related proteins [KBHR08, VMR+10]. However, methods largely based on this type of data are often limited by the same aspects that also compromise function prediction approaches, i.e. data quality. Thus, reliable data are also essential for successful disease gene identification.

To demonstrate the impact of data coverage and quality on disease gene identification we apply a network-based algorithm on two different human protein interaction networks: a sparse network (13,494 proteins and 43,637 interactions) and a more dense network (14,218 proteins and 81,868 interactions) generated with PiPa. Our approach first extracts for a given disease all proteins known to be associated with this disease. Based on these proteins we build a disease-specific network by integrating directly and indirectly interacting gene products. Proteins in this network are ranked based on network centrality and the most central proteins are considered to be highly relevant for the disease.

Figure 6 shows the cross-validation recovery rate obtained for each network. The comparison shows that using a network with a higher coverage increases the recovery rate of blinded disease proteins considerably. In contrast to the sparse network, the denser network allows to recover 52% of the disease proteins instead of 38%. The increase emphasizes the importance of coherent and complete interaction data for inferring disease gene associations.

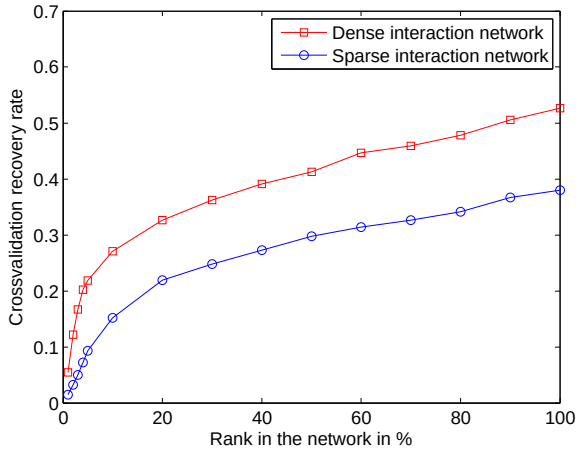

Figure 6: Impact of sparse and dense protein interaction data on the performance of disease gene identification.

5.3 Evaluation of text mining applications

Supporting evaluation strategies for text mining methods is another field of application for PiPa. Comprehensive data sets integrated by PiPa are very suitable to (1) estimate the quality of computational extraction of protein-protein interactions from text and to (2) unveil the state of the art in reconstructing biological pathways using Natural Language Processing (NLP).

Relation extraction is commonly performed by a relation-specific classifier based on machine learning techniques like support vector machines, pattern matching, etc. [ZH08]. Such techniques require a set of manually classified examples (training corpus) in order to train the classifier initially. Since the training corpus is only a small set of samples it is inherent biased compared to the application corpus [WKS+10]. Furthermore, supervised learning tends to overfit on the training data [TTP+10]. Both characteristics result in an overrated performance on the training corpus determined by e.g. cross-validation. Without an appropriate gold standard it is hardly possible to measure the real performance of such methods. Therefore, the diverse and rich PPI and pathway datasets provided by PiPa are able to fill that gap.

Since the integrated data originate from miscellaneous manually curated databases they can be considered as a realistic and qualitative gold standard suitable for a proper evaluation of text mining methods. In case of protein-protein interactions PiPa additionally integrates background information that, among others, contains references to PubMed articles providing evidence for each interaction. Such links enable the compilation of an appropriate evaluation corpus and its corresponding interactions.

Another application of PiPa comes up with the use of NLP for automatic reconstruction of biological pathways from text. For example Rodríguez-Penagos et al. [RPSMFCV07] present a rule-based system capable to directly generate regulatory networks of *Escherichia coli* from abstracts and full-text papers. By providing diverse pathway data PiPa enables the evaluation of such approaches. Comparing the integrated pathway data with computationally generated networks allows the detection of coverage and even may lead to novel extensions of known pathways.

6 Conclusion

PiPa is a highly modularized system for building local databases integrating a large number of original databases from the area of biological pathways, protein-protein-interactions, and protein-related information. The need for PiPa arose from a number of in-house projects that all depend on simple access to comprehensive PPI and pathway data. However, as the precise ways of analyzing this data vary greatly between projects, we deliberately kept PiPa's functionality strictly at what is necessary to provide a comprehensive and current data base for System Biology research. In addition, PiPa provides functionalities to control and monitor the update procedures of the different resources.

Its modular design makes PiPa flexible enough to be quickly customized to different in-

tegration needs. Available plug-ins (and thus import modules) can be removed from the system simply by changing configuration information, as virtually all modules are independent from each other; the only exception is the plug-in for UniProt that is necessary for all other plug-ins to perform ID normalization. Furthermore, new plug-ins for integrating new data sources can be added, building on the rich functionality provided by the PiPa framework.

However, there are also some aspects that need further improvements. For instance, although Java generally allows cross platform deployment, PiPa currently uses code specific to UNIX-style environments. This restriction will be removed in the future. Another issue are load times. The time it takes to import a source depends on its size and, in particular, on the number of proteins it references that are not yet contained in the database, because especially the step of UniProt ID resolution and data load is time-consuming (this implies that average loading times per item actually go down with increasing database content). However, we deliberately chose the dynamic data completion approach (see Section 4) to ensure that, whenever a source is loaded, the referenced information from UniProt has the same level of timeliness. From our experience, data loads are not performed too frequently, as many types of analysis require a stable data set during development.

References

[AJO06] R. Aragues, D. Jaeggi, and B. Oliva. PIANA: protein interactions and network analysis. *Bioinformatics*, 22(8):1015, 2006.

[BBC+05] G. Bader, E. Brauner, M. Cary, R. Goldberg, C. Hogue, P. Karp, T. Klein, J. Luciano, D. Marks, N. Maltsev, et al. BioPAX–Biological Pathways Exchange language. *BioPAX Workgroup*, 2005.

[BQG+06] Michael Baitaluk, Xufei Qian, Shubhada Godbole, Alpan Raval, Animesh Ray, and Amarnath Gupta. PathSys: integrating molecular interaction graphs for systems biology. *BMC bioinformatics*, 7(1), February 2006.

[CMH09] J. Chen, S.R. Mamidipalli, and T. Huan. HAPPI: an online database of comprehensive human annotated and predicted protein interactions. *BMC genomics*, 10(Suppl 1):S16, 2009.

[CMR+09] G. Chaurasia, S. Malhotra, J. Russ, S. Schnoegl, C. H
"anig, E.E. Wanker, and M.E. Futschik. UniHI 4: new tools for query, analysis and visualization of the human protein–protein interactome. *Nucleic acids research*, 37(suppl 1):D657, 2009.

[CSW07] Hon Nian Chua, Wing-Kin Sung, and Limsoon Wong. Using indirect protein interactions for the prediction of Gene Ontology functions. *BMC Bioinformatics*, 8 Suppl 4:S8, 2007.

[CWZ+09] S. Chowbina, X. Wu, F. Zhang, P. Li, R. Pandey, H. Kasamsetty, and J. Chen. HPD: an online integrated human pathway database enabling systems biology studies. *BMC bioinformatics*, 10(Suppl 11):S5, 2009.

[DKR+08] Marcus T Dittrich, Gunnar W Klau, Andreas Rosenwald, Thomas Dandekar, and Tobias Müller. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics*, 24(13):i223–i231, Jul 2008.

[EMXY00] D. Eisenberg, E.M. Marcotte, I. Xenarios, and T.O. Yeates. Protein function in the post-genomic era. *Nature*, 405(6788):823–826, 2000.

[FBA+08] O. Filangi, Y. Beausse, A. Assi, L. Legrand, J.M. Larré, V. Martin, O. Collin, C. Caron, H. Leroy, and D. Allouche. BioMAJ: a flexible framework for databanks synchronization and processing. *Bioinformatics*, 24(16):1823, 2008.

[GWPL08] P. Groth, B. Weiss, H.D. Pohlenz, and U. Leser. Mining phenotypes for gene function prediction. *BMC bioinformatics*, 9(1):136, 2008.

[HMPB+04] H. Hermjakob, L. Montecchi-Palazzi, G. Bader, J. Wojcik, L. Salwinski, A. Ceol, S. Moore, S. Orchard, U. Sarkans, C. Von Mering, et al. The HUPO PSI's molecular interaction formatâa community standard for the representation of protein interaction data. *Nature biotechnology*, 22(2):177–183, 2004.

[IS08] Trey Ideker and Roded Sharan. Protein networks in disease. *Genome Res*, 18(4):644–652, Apr 2008.

[JGLRS08] S. Jaeger, S. Gaudan, U. Leser, and D. Rebholz-Schuhmann. Integrating protein-protein interactions and text mining for protein function prediction. *BMC bioinformatics*, 9(Suppl 8):S2, 2008.

[JL07] Samira Jaeger and Ulf Leser. High-Precision Function Prediction using Conserved Interactions. In Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron, and Dirk Walther, editors, *Proceedings of the German Conference on Bioinformatics, GCB 2007, September 26-28, 2007, Potsdam, Germany*, volume 115 of *LNI*, pages 146–162. GI, 2007.

[JSL10] Samira Jaeger, Christine T Sers, and Ulf Leser. Combining modularity, conservation, and interactions of proteins significantly increases precision and coverage of protein function prediction. *BMC Genomics*, 11:717, 2010.

[KBHR08] Sebastian Köhler, Sebastian Bauer, Denise Horn, and Peter N Robinson. Walking the interactome for prioritization of candidate disease genes. *Am J Hum Genet*, 82(4):949–958, Apr 2008.

[KWLH09] A. Kamburov, C. Wierling, H. Lehrach, and R. Herwig. ConsensusPathDBâa database for integrating human functional interaction networks. *Nucleic acids research*, 37(suppl 1):D623, 2009.

[LS09] B. Lehne and T. Schlitt. Protein–protein interaction databases: Keeping up with growing interactomes. *Human genomics*, 3(3):291–297, 2009.

[PF95] E. M. Phizicky and S. Fields. Protein-protein interactions: methods for detection and analysis. *Microbiol Rev*, 59(1):94–123, Mar 1995.

[PNA+03] S. Peri, J.D. Navarro, R. Amanchy, T.Z. Kristiansen, C.K. Jonnalagadda, V. Surendranath, V. Niranjan, B. Muthusamy, TKB Gandhi, M. Gronborg, et al. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome research*, 13(10):2363, 2003.

[PWK+08] S. Patient, D. Wieser, M. Kleen, E. Kretschmann, M. Jesus Martin, and R. Apweiler. UniProtJAPI: a remote API for accessing UniProt data. *Bioinformatics*, 24(10):1321, 2008.

[RPSMFCV07] C. Rodríguez-Penagos, H. Salgado, I. Martínez-Flores, and J. Collado-Vides. Automatic reconstruction of a bacterial regulatory network using Natural Language Processing. *BMC bioinformatics*, 8(1):293, 2007.

[SAK+09] C.F. Schaefer, K. Anthony, S. Krupa, J. Buchoff, M. Day, T. Hannay, and K.H. Buetow. PID: the pathway interaction database. *Nucleic acids research*, 37(suppl 1):D674, 2009.

[Sch04] C.F. Schaefer. Pathway databases. *Annals of the New York Academy of Sciences*, 1020(1):77–91, 2004.

[SHX+05] S.P. Shah, Y. Huang, T. Xu, M. Yuen, J. Ling, and BF Ouellette. Atlas – a data warehouse for integrative bioinformatics. *BMC bioinformatics*, 6(1):34, 2005.

[SM03] Victor Spirin and Leonid A Mirny. Protein complexes and functional modules in molecular networks. *Proc Natl Acad Sci U S A*, 100(21):12123–12128, Oct 2003.

[SSK+05] Roded Sharan, Silpa Suthram, Ryan M Kelley, Tanja Kuhn, Scott McCuine, Peter Uetz, Taylor Sittler, Richard M Karp, and Trey Ideker. Conserved patterns of protein interaction in multiple species. *Proc Natl Acad Sci U S A*, 102(6):1974–1979, Feb 2005.

[SUS07] Roded Sharan, Igor Ulitsky, and Ron Shamir. Network-based prediction of protein function. *Mol Syst Biol*, 3:88, 2007.

[TTP+10] D. Tikk, P. Thomas, P. Palaga, J. Hakenberg, and U. Leser. A Comprehensive Benchmark of Kernel Methods to Extract Protein–Protein Interactions from Literature. *PLoS Computational Biology*, 6(7):e1000837, 2010.

[VFMV03] Alexei Vazquez, Alessandro Flammini, Amos Maritan, and Alessandro Vespignani. Global protein function prediction from protein-protein interaction networks. *Nat Biotechnol*, 21(6):697–700, Jun 2003.

[VMR+10] Oron Vanunu, Oded Magger, Eytan Ruppin, Tomer Shlomi, and Roded Sharan. Associating genes and protein complexes with disease via network propagation. *PLoS Comput Biol*, 6(1):e1000641, Jan 2010.

[WKS+10] Y. Wang, J.I.N.D. KIM, R. Saetre, S. Pyysalo, T. Ohta, and T. JUN'ICHI. Improving the inter-corpora compatibility for protein annotations. *Journal of bioinformatics and computational biology*, 8(5):901–916, 2010.

[ZH08] D. Zhou and Y. He. Extracting interactions between proteins from the literature. *Journal of biomedical informatics*, 41(2):393–407, 2008.