BN++ - A Biological Information System

Jan Küntzer[1], Torsten Blum[2], Andreas Gerasch[2], Christina Backes[1], Andreas Hildebrandt[1], Michael Kaufmann[2], Oliver Kohlbacher[2], Hans-Peter Lenhof[1]

[1]Center for Bioinformatics, Saarland University, 66041 Saarbrücken, Germany

[2]Center for Bioinformatics/Wilhelm Schickard Institute for Computer Science,
Eberhard-Karls-Universität Tübingen, 72076 Tübingen, Germany

Summary

Recent years have seen an explosive growth in the amount of biochemical data available. Numerous databases have been established and are being used as an essential resource by biologists around the world. The sheer amount and heterogeneity of these data poses a major challenge: data integration and, based thereupon, the integrative analysis of these data. We present BN++, the biochemical network library, a powerful software package for integrating, analyzing, and visualizing biochemical data in the context of networks. BN++ is based on a comprehensive and extensible object model (BioCore), which has been implemented as a C++ framework, a Java class library, and a relational database. The C++ framework is used to efficiently import, integrate, and analyze the data, which is stored in a data warehouse. The Java-based viewer (BiNA) provides a powerful platform-independent visualization of the data using sophisticated graph layout algorithms. Currently, the data warehouse imports and integrates data from about a dozen important databases including, among others, sequence data, metabolic and regulatory networks, and protein interaction data. We illustrate the usefulness of BN++ with a few select example applications.
Availability: BN++ is open source software available from our website at `www.bnplusplus.org`.

1 Introduction

Systems biology has seen an explosion in the amount of data available lately. This growth is mainly caused by novel high-throughput techniques. A wealth of data are collected by an ever increasing amount of databases [1] and made available mostly through web-servers or as flat text files. One of the key challenges in computational systems biology is thus the integration of large heterogeneous data sets and the analysis of these data in an integrated fashion. Hence there is a pressing need for more efficient systems for integrating, analyzing, and interpreting these data jointly. A large number of software systems addressing these issues have been developed over the last decade. These approaches can be classified by their architecture into three main categories [2]: *navigators*, *mediators*, and *warehouses*. The first category, navigators, consists mainly of link-driven, web-based schemes offering an interface to navigate and search by keywords through several data sources. Such a portal normally does not integrate the data itself. Examples for portal systems are SRS [3], BioNavigator [4], and Entrez [5]. The second category allows access to distributed data using mediators, which are wrappers to translate a query at runtime into the scheme of the external databases. Hence Mediators do not require local databases, and circumvent the problems of possibly outdated results. Examples for this category are Discovery Link [6], TAMBIS [7], and BioMediator [8]. A data warehouse, the third category, relies on complete semantic integration of data from various external sources into a local database. This approach permits direct access to the database and enabling efficient query optimization and execution, especially for huge datasets. Another benefit is the possibility to add own data into a warehouse system. A disadvantage is the high complexity in the integration process as well as the need for regular data updates. Examples for this approach are GUS [9] and Biozon [10]. These three fundamental categories can be combined into hybrid systems as well. An example for these hybrid systems is MARGBench [11], which combines features of a warehouse system with additional mediators.

In this work we present BN++, an integrated software package for computational systems biology consisting of the following closely connected components:

- an object-oriented data model (BioCore) for the representation of biochemical data and processes,
- a C++ and a Java implementation of this object model,
- a data warehouse integrating a large number of important data sources, and
- a graphical user-interface (BiNA) for visualizing, navigating, and analyzing these data.

At the heart of this software system is BioCore, the object-oriented data model. It is powerful enough to model most known biochemical processes and at the same time easily extensible to be adapted to new biological concepts.

BN++ was designed to be usable by both, software developers and biologically oriented users. For software developers, we provide two implementations of the data model (in C++ and Java). Software developers will appreciate the rapid software prototyping features that result in short turn-over times during application development. In contrast, users whose primary concern lies with a specific biological problem will prefer the convenient graphical user interface BiNA. The interface allows complex queries to the data warehouse and conveniently visualizes the results of these queries using automated graph drawing. A flexible plugin structure allows the users to add extra functionality easily.

We have implemented a broad range of importers for widely used databases (RefSeq [12], KEGG [13], BioCyc [14], TransPath [15], among others) and for standard data exchange formats (e.g. PSI-MIF level 1[16]). The integration of these data requires complex merging heuristics that have been implemented for some of the key data sources.

The following section introduces the design and the architecture of BN++ and presents the data model, the framework, the database and the graphical user interface in detail. A few biological applications and their implementations with BN++ are presented in Section 3. In the final section of this paper we discuss the assets and drawbacks of our approach and give an outlook on further developments.

2 Design and Architecture

Applications in systems biology require powerful, yet easy to use tools with a high degree of flexibility. The knowledge of biochemical processes and mechanisms is growing rapidly, inducing a continuous upgrading and expanding of biochemical process models. Therefore the extensibility of the integrative system is one of its essential properties. The above mentioned goals – ease-of-use, rich functionality, and extensibility – are partially conflicting and thus careful design and architecture of the system as a whole is of the utmost importance. In this section we will sketch the design of our solution to this challenge.

The overall architecture of BN++ is presented in Fig. 1. The heart of BN++ is a comprehensive data model called BioCore that allows to model almost all biochemical processes. In addition, for users familiar with the UML design and the object-oriented concept, the model can be easily extended to account for novel biochemical concepts and mechanisms. Starting from a UML model of BioCore, we have implemented a C++ framework for rapid application development as well as an equivalent Java class library. Both frameworks enable the users to realize new applications efficiently with a well-tested code basis.

Based on the BioCore data model we have developed and implemented an SQL data warehouse system that integrates data sets from external data sources via importers. Already in the current state the data warehouse represents a comprehensive collection of data integrated from the following external and internal sources:

- Sequence databases: SwissProt [17], RefSeq [12]
- Pathway databases: KEGG [13], BioCyc [14], TransPath [15]
- Protein interaction databases: DIP [18], MINT [19], IntAct [20], HPRD [21]
- Transcription factor databases: TransFac [22]
- Protein annotation databases: InterPro [23], CAP [24]

Based on the Java class library of BN++ we developed BiNA, a graphical user interface (GUI) and network visualizer that enables the users to carry out complex queries of the BN++ warehouse using an intuitive interface without requiring any knowledge about BioCore, the database internals, or SQL. Using sophisticated graph layout algorithms query results can be easily visualized as graphs or networks in a visually appealing manner. This greatly enhances the usability for those who want to apply BN++ to their own field of research.

In addition to the rich functionality already realized in BiNA, the application is easily extended through a plugin interface similar to the plugin structure of Cytoscape [25], albeit more powerful. Through this plugin interface one could easily add analysis and layout algorithms to BiNA (see for example the ScorePAGE algorithm discussed in Section 3).

2.1 Data model

At the core of the biochemical network library is its comprehensive data model (BioCore), which has been developed using the Unified Modeling Language [26]. A previous version of BioCore was presented in [27]. The model contains more than 200 classes, allowing to represent most currently known biochemical entities and processes, but also offers an easy extensibility. The model is centered around three fundamental classes `Event`, `Role`, and `Participant` (see Fig. 2).

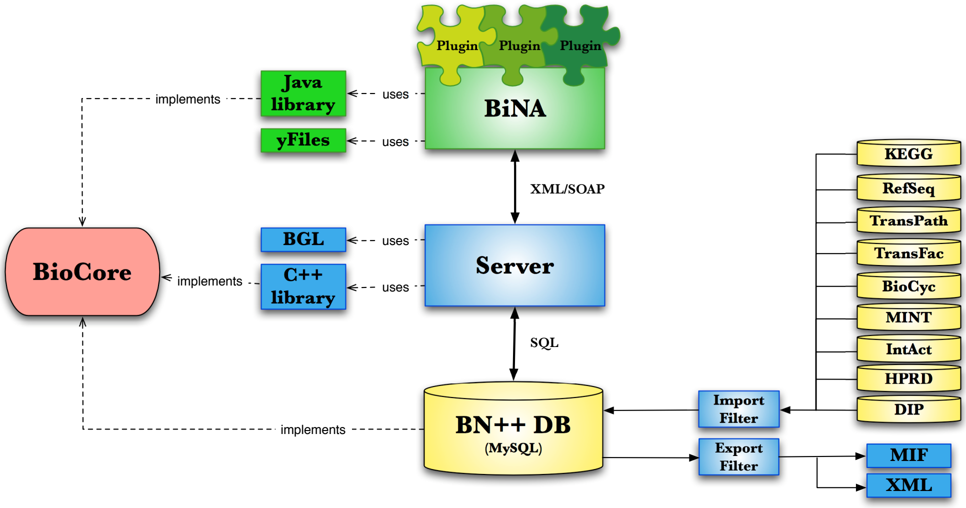


Figure 1: Architecture of BN++

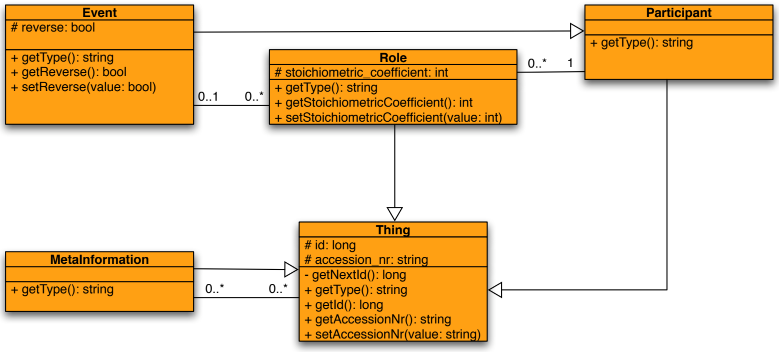


Figure 2: UML diagram of the central kernel classes in the biochemical network library (simplified).

Biochemical processes are modeled as `Events` with `Participants` playing a certain `Role`. BioCore contains a large number of predefined `Participant` classes (Gene, DNA, RNA, Protein, Compound, etc.), `Role` classes (Product, Educt, Enzyme, Inhibitor, Enhancer, etc.), and Event classes (Reaction, Interaction, Expression, Translation, Splicing, etc.). As an example we present the modelling of an enzymatic reaction (see Fig. 3).

A user can easily extend the functionality by subclassing from the core classes thus introducing novel molecular processes or mechanisms. This is demonstrated for the case of the gene silencing, a gene regulation process, that has recently received considerable attention in the literature (see Figure 4).

The data model has been implemented in C++, Java and as a relational database. We will now briefly describe these implementations.

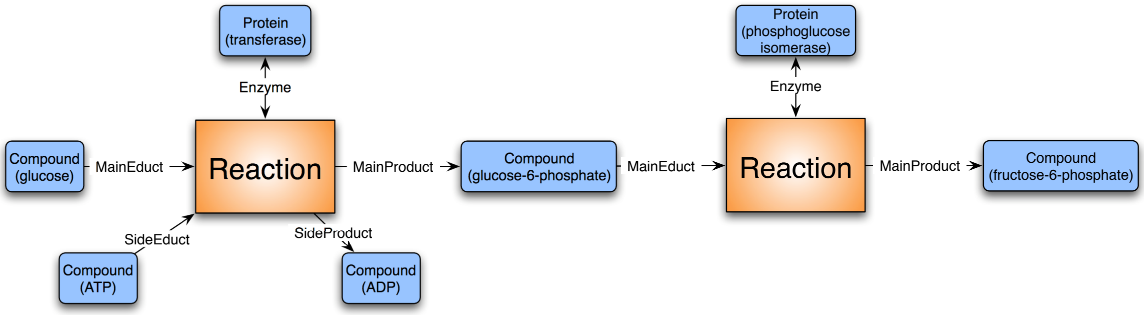


Figure 3: Modeling of the first two steps in the glycolysis as enzymatic reactions in BioCore. Blue boxes stand for participants, orange boxes for events, and the edges (arrows) are labeled with the roles the participants play in the particular reaction. For clarity, each participant is labeled with its name (shown in brackets).

2.2 C++ implementation

The C++ implementation of the BioCore model is a significantly revised and extended version of the implementation previously presented in [27]. It implements a one-to-one mapping of the BioCore classes onto C++ classes of the same name. These classes form the kernel of the C++ framework, which is further extended by additional classes providing functionality for data integration as well as for data analysis. For data import, the C++ framework offers importer and exporter capabilities for a number of different data sources (see below). The concept behind data importers is based on the mapping of the external data source model onto BioCore objects: Each biochemical process needs to be mapped onto a corresponding event class. Furthermore, all participants and the roles they play in the event, have to be instantiated. All additional information need to be mapped onto suitable metainformation instances. Such an importer returns an object connected with all the instances, e.g., a data source object. It is then possible to serialize this collection of objects (e.g. to the database) with a single line of code. An implementation of the Molecular Interaction Format MIF Level 1.0 [16] by the proteomics standard initiative provides the means to integrate numerous datasources from the field of protein-protein interactions [19, 18, 20, 21].

In order to exchange objects in a platform independent manner the C++ framework also provides full support for the Simple Object Access Protocol (SOAP) [28]. SOAP offers a way to communicate between applications running on different operating systems with different technologies and programming languages. BN++ provides a WSDL-file (Web Services Description Language) [29], that describes the web service and specifies the location of the service and the methods the service exposes. It can be used to easily access BN++ applications from all common programming languages through a single interface.

Mathematical graph representations are often the method of choice for the analysis of complex biochemical data in the context of networks. However, there is no unique mapping of biochemical networks onto a single graph structure. We thus provide a generic mapping, that allows us to map arbitrary BioCore classes onto the nodes and edges of a graph. For example mapping the class `Protein` onto the nodes and the class `Interaction` onto the edges realizes a typical protein interaction graph. Similarly by mapping enzymes and metabolites onto nodes and their respective roles in metabolic reactions onto the edges, results in a metabolic network. These

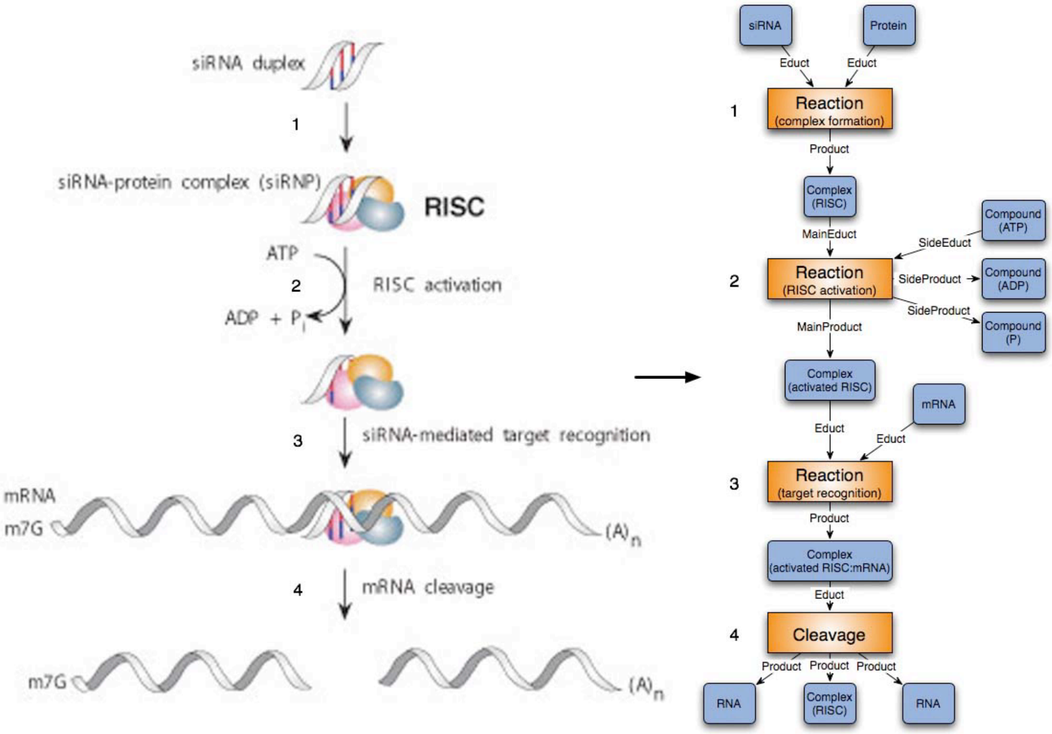


Figure 4: Examplary modeling of gene silencing through RNA interference. The left-hand side shows the biological mechanism starting with the activation of RISC (RNA-induced silencing complex) by associaion with siRNA, unwinding of the duplex and removal of one of the strands (with kind permission from Cell Signaling Technology, www.cellsignal.com). The right-hand side of the figure illustrates a detailed modeling of this process with BioCore. Blue boxes stand for participants, orange boxes for events, and the edges (arrows) are labeled with the roles the participants play in the particular reaction. Participants and events are labeled with their names (shown in brackets).

graphs can further be augmented by coloring their nodes or edges with respect to auxiliary data (e.g. expression data, concentrations). The implementation of the graph data structure is based on the boost graph library (BGL) [30]. BGL also provides a number of graph algorithms like shortest paths, minimum spanning trees, connected components, etc.

2.3 Database

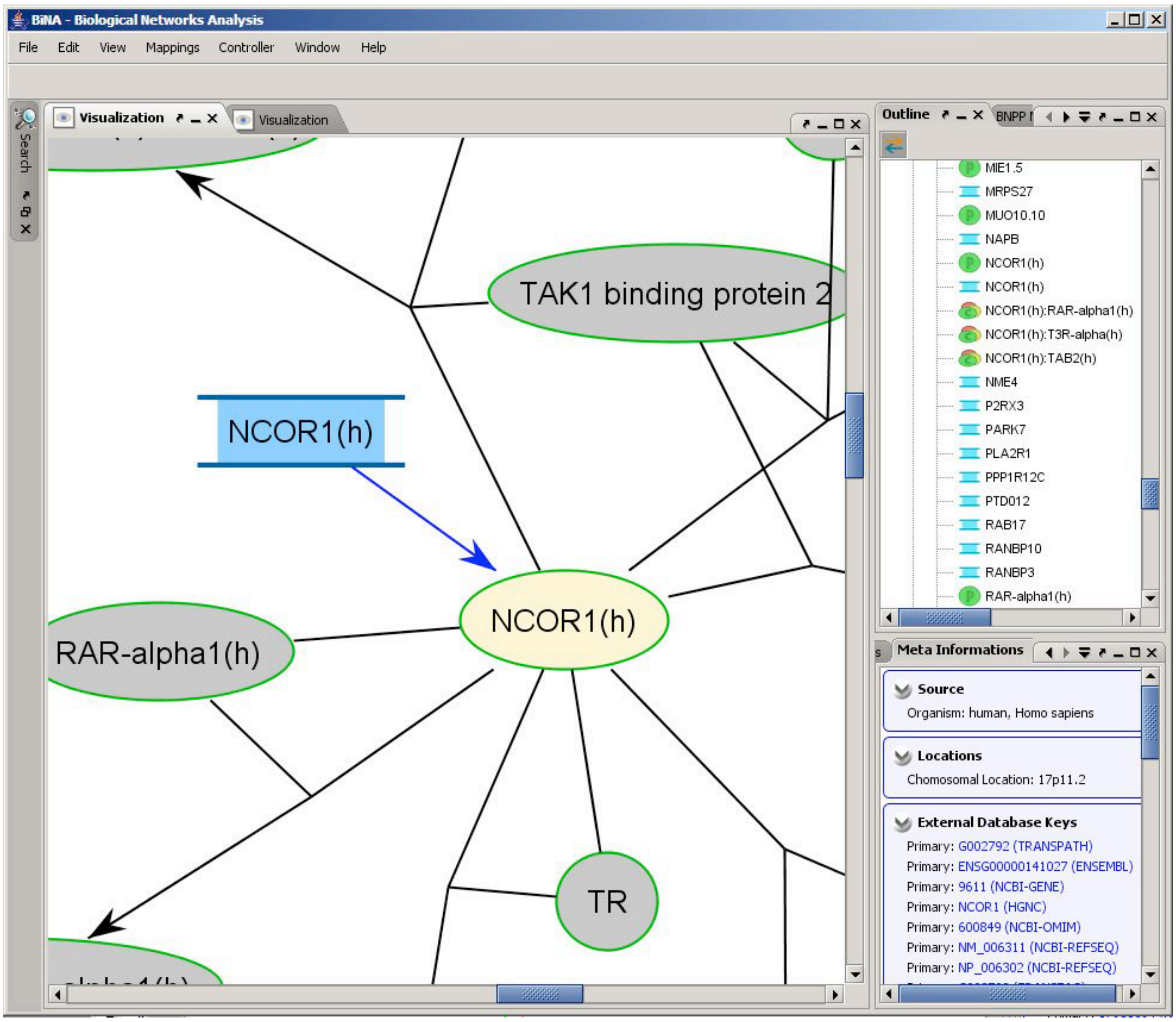
At the heart of our data warehouse system lies a relational database. We chose a relational database system over an object-oriented architecture, because relational databases are well-established and the current de-facto standard. In particular, the availability of relational database management systems for a wide range of platforms ensures a high portability. We created an object-relational mapping of the BioCore model onto relational database management systems (DBMS). This model can be easily realized in any relational or object-relational DBMS like DB2, PostgreSQL, or MySQL. Our current implementation uses MySQL, nevertheless a deliberate restriction to SQL2 [31] compatible statements ensures a high degree of portability.

The tight integration of the C++ framework with the database is useful when addressing horizontal data integration. The key objective here is the integration of complementary data sources and the elimination of redundancy in the data. Currently, there is no generally applicable solution to this problem. Merging multiple databases still requires custom-built heuristics and often enough manual curation of the resulting merged databases. In order to facilitate the merging, BN++ provides convenient methods for merging BioCore object across databases with just a few lines of code. All names, descriptions, and data source annotations can be automatically combined using several hand-coded heuristics. For a number of databases (e.g. RefSeq, Swiss-Prot, and KEGG) we have implemented automatic mergers employing external data (e.g. the NCBI taxonomy database [32]) to carefully unify the data. The heuristics rely on the existence and correctness of selected standardized ids in the imported databases. Each object in the database is linked with a variety of different external datasource identifiers. We use these ids to unify the data, where the selection of the identifiers needs to be done carefully. Some of the external database ids are not describing unique objects, but rather clusters. After selecting the databases with unambiguous ids, all objects associated with identical identifiers are detected and merged using the C++ framework. For example the organisms are automatically unified and named by means of NCBI taxonomy identifiers. In addition we define two events to be mergable, if they are of the same type and connected with the same participants using the same roles. These events are detected and merged in the database.Administration and manual searching or editing of the database is simplified by a simple web interface.

2.4 Visualization with BiNA

Navigating the wealth of data contained in a large data warehouse and analyzing these data conveniently is a major challenge. Our approach to this problem is a Java-based visualization tool (BiNA, *Biological Network Analysis tool*). BiNA serves both, as a front end to the data warehouse and as a highly sophisticated visualization tool for biochemical network data. At the core of BiNA lies a generic visualization of network data, which provides concise representations for the different levels of biological networks: metabolic, regulatory, and interaction networks. These types of networks can be manually rearranged and dynamically navigated by the user. The direct connection of the viewer to the data warehouse allows the direct retrieval of arbitrary meta information related to any of the objects displayed. A prominent feature of BiNA is the displaying of multiple layout styles in the same graph view. This can be done by dividing the displayed graph into groups and defining different layout styles. The supported styles on the methodological sides [33, 34] are organical, hierarchical, orthogonal and tree-like as provided by the yFiles library. Automatic or manually curated layouts of partial networks can be stored in the database or retrieved from there (a screenshot is presented in Figure 5).

BiNA allows not only the navigation and visualization of complex data, but also its analysis. After loading a data set (e.g. biological network) from the database into BiNA, it is possible to extend the graph by dynamically reloading associated biological events via a breadth-first or a shortest-path-to search. Restricting this search to subsets of interest can be done by defining meta information filters, such as `everything but` ... and `nothing but` ...- filter. Useful meta information associated with objects are organisms and source databases, such that biological events in different organisms can be compared easily. Besides the direct visualization and comparison of networks, BiNA also provides a mapping engine to analyze arbitrary datasets in the context of networks. It is possible to map arbitrary scalar data onto graph attributes by changing node/edge visibility, color, size and shape, position, or layout style. This can be done either using one of several convenient predefined mappings (e.g. gene expression mapping) or through the BiNA plugin system (a screenshot of the mapping dialog is presented



Source

Organism: human, Homo sapiens

Locations

Chomosomal Location: 17p11.2

External Database Keys

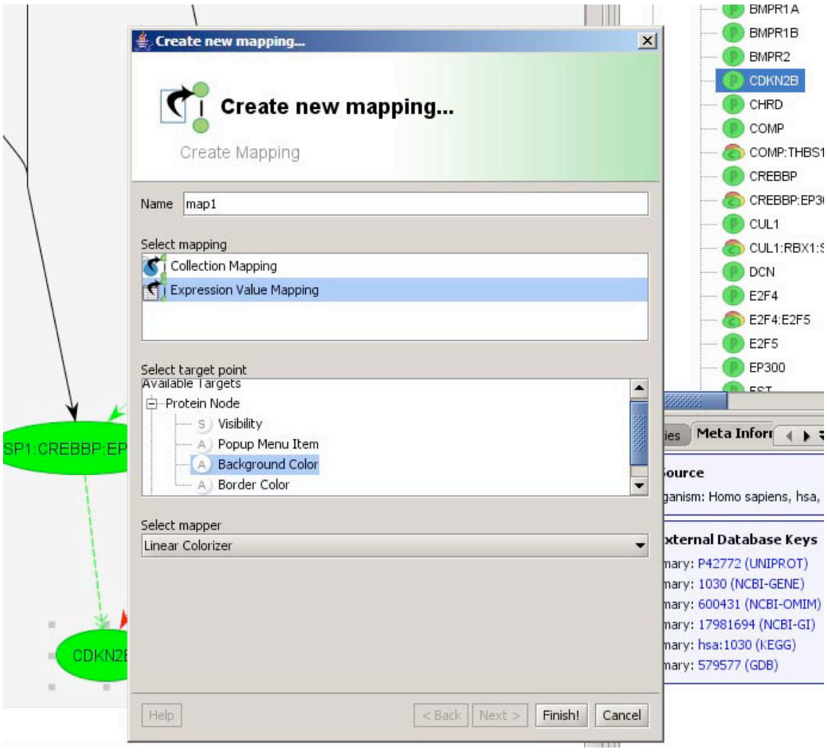
Primary: G002792 (TRANSPATH)
Primary: ENSG00000141027 (ENSEMBL)
Primary: 9611 (NCBI-GENE)
Primary: NCOR1 (HGNC)
Primary: 600849 (NCBI-OMIM)
Primary: NM_006311 (NCBI-REFSEQ)
Primary: NP_006302 (NCBI-REFSEQ)

Figure 5: The human TGF-beta signaling pathway visualized with the regulatory view in BiNA. Each participant is shown as a node labeled with its name. The blue box represents a gene and the ovals proteins. The events are visualized with different types of edges: the black arrows represent different reactions, the blue arrow an expression.

in Figure 6).

BiNA is easily extensible through a powerful plugin system. The viewer itself can be regarded as a collection of modules that depend on each other. The hierarchical plugin system automatically resolves dependencies between individual plugins. Thus, plugins can extend other plugins through a well-defined interface. As the full details of this plugin architecture is beyond the scope of this work, we will only give two examples of possible extensions. Plugins can interfere at a very basic level even with the core functionality of BiNA. Thus, it is possible to provide new graph layout algorithms. An example for this are fish-eye layouts for protein interaction graphs [35] that are easily implemented as a plugin. It is also possible to provide plugins for specific import or export formats or new analysis algorithms (see the ScorePAGE algorithm in Section 3).

The implementation of BiNA relies on the yFiles graph-visualization library [36] and its excellent layout algorithms. This Java library is one of the best-established libraries in the field of graph drawing and is used in various applications going far beyond the field of visualization of



Create new mapping...

Create Mapping

Name: map1

Select mapping: Expression Value Mapping

Select target point: Protein Node – Background Color

Select mapper: Linear Colorizer

Figure 6: This dialog defines a new mapping of expression values on the background color of protein nodes.

biological networks. Comparable functionality is only offered by the high-prized graph layout software of TomSawyer [1]. Combining the features of the several components of BiNA (BN++ data model, visualization, etc.) with the flexible graphical user interface and the powerful plugin management system of the BiNA platform, we have an easily extensible multifunctional workbench available.

3 Applications

The biochemical network library can be applied to wide variety of interesting problems. Here, we briefly discuss a few select examples of biological applications of BN++.

The first application we realized on top of BN++, a program for finding metabolic pathways in newly sequenced organisms, has been presented in [27]. The algorithm, called PathFinder, tries to identify the metabolic pathways of the unknown organism from the genomic sequence of a newly sequenced organism and the metabolic pathways of a set of closely related reference organisms. BN++ can also be used to compare and annotate metabolic networks. By mapping the merged data from different datasources onto one consistent graph, we are able to identify differences and omissions in databases. Comparing these graphs across species allows us to close gaps (e.g. yet unannotated enzymes) in metabolic networks.

As a second application we developed the web-tool GeneTrail for the analysis of certain properties of gene sets. The web-tool provides the possibility to analyze different gene sets in

[1] http://www.tomsawyer.com

comparison to a chosen reference set. For example, GeneTrail can be employed whether genes in a given set are enriched/overrepresented or underrepresented on certain metabolic or regulatory pathways. This analysis is based along the biochemical network data stored in the BN++ data warehouse.

Another project that has been integrated in BN++ is the cancer-associated protein database (CAP), an integrative analysis system for cancer-related data. In [24] we have employed CAP to analyze genes (tumor antigens) that have been found to cause an autoimmune response in cancer. In particular, we explored the connection between the autoimmune response, mutations, and overexpression of the genes. Our preliminary results indicate that mutations are not significant contributors to raising an antibody response against tumor antigens, whereas overexpression seems to play a more important role. Integrative analysis of this type is greatly simplified by the horizontal data integration provided by BN++.

An example for the extension of BiNA through its plugin structure is the implementation of the ScorePAGE (Scoring Pathway Activity from Gene Expression) algorithm by Rahnenführer et. al. [37]. ScorePAGE identifies activity changes in metabolic pathways. The algorithm combines metabolic pathway information with expression data in a novel topological scoring function to evaluate pathway activity. The scores depend on all genes coding for enzymes on a given pathway, where pairwise co-regulation of these genes is taken into account. Including the biological knowledge of pathway data in this way, even subtle changes in the activity can be detected. The significance of co-regulation is calculated from a nonparametric permutation test, randomly permuting gene label assignments. Since we provide a BiNA plugin for the ScorePAGE algorithm, the user can easily choose arbitrary expression experiments as well as a gene set. The ScorePAGE plugin searches for all pathways containing at least one gene from the selected set. The user can choose an arbitrary gene set in BiNA by selecting nodes of metabolic pathways. In addition, the ScorePAGE plugin provides the option of choosing between one or more expression experiments contained in the BN++ database. The result of the ScorePAGE analysis lists all these pathways along with their computed scores and their statistical significance values.

4 Discussion and Outlook

With the biochemical network library BN++ we present a novel software system for computational systems biology. BN++ is centered around a comprehensive integrative object-oriented data model. The models expressivity allows to model all relevant biological processes accurately, while being intuitive enough even for less experienced users. Implementations of this object model in both, C++ and Java provide convenient means for rapid software prototyping of complex applications in systems biology.

BN++ is targeted at two distinct groups of user. The first group are bioinformaticians with a basic knowledge of C++, who want to use BN++ as a rapid prototyping library. This is facilitated by BN++'s rich functionality and its easy-of-use. Through its graph library, BN++ offers the possibility to analyze complex data in the context of biological networks with little effort. The second category of users are biologists using either the data warehouse to have unified access to their most important data sources or BiNA as a visualization tool. The graph and visualization capabilities of our application are comparable to that of visualization systems such as Cytoscape [25], PathSys [38] or VisANT [39] as well as commercial tools such as MetaDrug [40] or PathwayStudio [41]. In contrast to these, BiNA offers a multifunctional workbench, which is easily extensible in every direction. For example the mapping of expression data onto the graph provides for users of this category the possibility to easily grasp relations.

Currently one of the major shortcoming of BN++ is the lack of an update concept for the database integrated. This clearly needs to be addressed in the future. The complex heuristics for data merging currently implemented in BN++ work well in practice. Nevertheless manual and automatic validation and curation of the merged data will be a mayor topic of future development.

Another focus of future efforts will be the implementation of additional standard file formats like MIF level 2.5 [16], BioPAX (`http://www.biopax.org`) and SBML [42]. This includes full export as well as import functionality for continuous development of additional database importer. Currently we are working on importers for BRENDA, GEO etc. Future versions of BiNA will also contain enhanced layout algorithms for regulatory and metabolic networks. A 2.5D view consisting of multiple layers representing the same network will offer a possibility to map, e.g., time series data onto biological networks.

5 Acknowledgements

The authors would like to thank Thomas Schäfer and Marite Sirava for their work on the BN++ project. We are further grateful to Dr. Jörg Rahnenführer for kindly providing us with the source of the ScorePAGE algorithm.

References

[1] M Y Galperin. The molecular biology database collection: 2006 update. *Nucleic Acids Res.*, 34:D3–D5, 2006. Database Issue.

[2] T Hernandez and S Kambhampati. Integration of biological sources: Current systems and challenges ahead. *SIGMOD Rec.*, 33(3):51–60, 2004.

[3] T Etzold and P Argos. SRS - an indexing and retrieval tool for flat file data libraries. *Comput. Appl. Biosci.*, 9(1):49–57, 1993.

[4] Entigen. BioNavigator - BioNode & BioNodeSA: Overview. `http://www.antigen.com/library`, 2001.

[5] National Center for Biotechnology Information. Entrez - Search and Retrieval System. `http://www.ncbi-nlm.nih.gov/Entrez`, 2006.

[6] L M Haas, P M Schwarz, P Kodali, E Kotlar, J E Rice, and W C Swope. DiscoveryLink: A system for integrated access to life sciences data sources. *IBM Systems Journal*, 40(2):489–511, 2001.

[7] R Stevens, P Baker, S Bechhofer, G Ng, A Jacoby, N W Paton, C A Goble, and A Brass. TAMBIS: transparent access to multiple bioinformatics information sources. *Bioinformatics*, 16(2):184–185, 2000.

[8] L Donelson, P Tarczy-Hornoch, P Mork, C Dolan, J A Mitchell, M Barrier, and H Mei. The BioMediator system as a data integration tool to answer diverse biologic queries. *Medinfo*, 11(2):768–772, 2004.

[9] S B Davidson, J Crabtree, B P Brunk, J Schug, V Tannen, G C Overton, and C J Stoeckert. K2/Kleisli and GUS: Experiments in integrated access to genomic data sources. *IBM Systems Journal*, 40(2):512–530, 2001.

[10] A Birkland and G Yona. BIOZON: a system for unification, management and analysis of heterogeneous biological data. *BMC Bioinformatics*, 7(70), 2006.

[11] A Freier, R Hofestädt, M Lange, and U Scholz. MARGBench - An Approach for Integration, Modeling and Animation of Metabolic Networks. pages 190–194. German Conference on Bioinformatics, 1999.

[12] K D Pruitt, T Tatusova, and D R Maglott. NCBI reference sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res.*, 33(1):D501–D504, 2005. Database Issue.

[13] M Kanehisa, S Goto, M Hattori, K F Aoki-Kinoshita, M Itoh, S Kawashima, T Katayama, M Araki, and M Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res.*, 34:D354–D357, 2006. Database Issue.

[14] C J Krieger, P Zhang, L A Mueller, A Wang, S Paley, M Arnaud, J Pick, S Y Rhee, and P D Karp. MetaCyc: a multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Res.*, 32(1):D438–D442, 2004. Database Issue.

[15] M Krull, S Pistor, N Voss, A Kel, I Reuter, D Kronenberg, H Michael, K Schwarzer, A Potapov, C Choi, O Kel-Margoulis, and E Wingender. TRANSPATH: An Information Resource for Storing and Visualizing Signaling Pathways and their Pathological Aberrations. *Nucleic Acids Res.*, 34:D546–D551, 2006. Database Issue.

[16] H Hermjakob, L Montecchi-Palazzi, G Bader, J Wojcik, L Salwinski, A Ceol, S Moore, S Orchard, U Sarkans, C von Mering, B Roechert, S Poux, E Jung, H Mersch, P Kersey, M Lappe, Y Lix, R Zeng, D Rana, M Nikolski, H Husi, C Brun, K Shanker, S G N Grant, C Sander, P Bork, W Zhu, A Pandey, A Brazma, B Jacq, M Vidal, D Sherman, P Legrain, G Cesareni, I Xenarios, D Eisenberg, B Steipe, C Hogue, and R Apweiler. The HUPO PSI Molecular Interaction Format - A community standard for the representation of protein interaction data. *Nat. Biotechnol.*, 22:177–183, 2004.

[17] C H Wu, R Apweiler, A Bairoch, D A Natale, W C Barker, B Boeckmann, S Ferro, E Gasteiger, H Huang, R Lopez, M Magrane, M J Martin, R Mazumder, C O'Donovan, N Redaschi, and B Suzek. The Universal Protein Resource (UniProt): an expanding universe of protein information. *Nucleic Acids Res.*, 34:D187–D191, 2006. Database Issue.

[18] L Salwinski, C S Miller, A J Smith, F K Pettit, J U Bowie, and D Eisenberg. The database of interacting proteins: 2004 update. *Nucleic Acids Res.*, 32:D449–D451, 2004. Database Issue.

[19] A Zanzoni, L Montecchi-Palazzi, M Quondam, G Ausiello, M Helmer-Citterich, and Cesareni G. MINT: a Molecular INTeraction database. *FEBS Letters*, 513(1):135–140, 2002.

[20] H Hermjakob, L Montecchi-Palazzi, C Lewington, S Mudali, S Kerrien, S Orchard, M Vingron, B Roechert, P Roepstorff, A Valencia, H Margalit, J Armstrong, A Bairoch, G Cesareni, D Sherman, and R Apweiler. IntAct - an open source molecular interaction database. *Nucleic Acids Res.*, 32:D452–D455, 2004. Database Issue.

[21] S Peri et al. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Research*, 13:2363–2371, 2003.

[22] V Matys, O Kel-Margoulis, E Fricke, I Liebich, S Land, A Barre-Dirrie, I Reuter, D Chekmenev, M Krull, K Hornischer, N Voss, P Stegmaier, B Lewicki-Potapov, H Saxel, A Kel, and E Wingender. TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res.*, 34:D108–D110, 2006. Database Issue.

[23] N J Mulder, R Apweiler, T K Attwood, A Bairoch, A Bateman, D Binns, P Bradley, P Bork, P Bucher, L Cerutti, R Copley, E Courcelle, U Das, R Durbin, W Fleischmann, J Gough, D Haft, N Harte, N Hulo, D Kahn, A Kanapin, M Krestyaninova, D Lonsdale, R Lopez, I Letunic, M Madera, J Maslen, J McDowall, A Mitchell, A N Nikolskaya, S Orchard, M Pagni, C P Ponting, E Quevillon, J Selengut, C J Sigrist, V Silventoinen, D J Studholme, R Vaughan, and C H Wu. InterPro, progress and status in 2005. *Nucleic Acids Res.*, 33:D201–D205, 2005. Database Issue.

[24] P Dönnes, A Höglund, M Sturm, N Comtesse, C Backes, E Meese, O Kohlbacher, and H P Lenhof. Integrative analysis of cancer-related data using CAP. *FASEB J.*, 18(12):1465–1467, 2004.

[25] P Shannon, A Markiel, O Ozier, N S Baliga, J T Wang, D Ramage, N Amin, B Schwikowski, and T Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, 2003.

[26] G Booch et al. *The Unified Modelling Language User Guide*. Addison Wesley Professional, 1999.

[27] M Sirava, T Schäfer, M Eigelsperger, O Kohlbacher, E Bornberg-Bauer, and H P Lenhof. BioMiner - modeling, analyzing, and visualizing biochemical pathways and networks. *Bioinformatics*, 18(2):219–230, 2002. http://www.zbi.uni-saarland.de/chair/projects/BioMiner.

[28] D Box. Simple Object Access Protocol (SOAP) 1.1. Technical report, World Wide Web Consortium (W3C), `http://www.w3.org/TR/SOAP`, 2000.

[29] E Christensen, F Curbera, G Meredith, and S Weerawarana. Web Services Description Language (WSDL) 1.1. Technical report, World Wide Web Consortium (W3C), `http://www.w3.org/TR/WSDL`, 2001.

[30] L Q Lee, A Lumsdaine, and J G Siek. *Boost Graph Library, The: User Guide and Reference Manual*. Addison Wesley Professional, 1st edition, 2001.

[31] ISO/IEC JTC1/SC21. Information Technology - Database Languages - SQL2. Technical report, ANSI, 1992.

[32] National Center for Biotechnology Information. The NCBI Taxonomy Browser. http://www.ncbi.nlm.nih.gov/Taxonomy, 2006.

[33] G Di Battista, P Eades, R Tamassia, and I G Tollis. Algorithms for drawing graphs: an annotated bibliography. *Comput. Geom. Theory Appl.*, 4(5):235–282, 1994.

[34] Kaufmann M and Wagner D. Drawing graphs: Methods and models. In *Lecture Notes in Computer Science*, 2025. Springer Verlag, 2001.

[35] E R Gansner, Y Koren, and S C North. Topological Fisheye Views for Visualizing Large Graphs. *IEEE Trans. Vis. Comput. Graph*, 11(4):457–468, 2005.

[36] R Wiese, M Eiglsperger, and M Kaufmann. yFiles: Visualization and Automatic Layout of Graphs. In *11th Symposium on Graph Drawing (GD'01)*. LNCS, 2001.

[37] J Rahnenführer, F S Domingues, J Maydt, and T Lengauer. Calculating the statistical significance of changes in pathway activity from gene expression data. *Stat. Appl. Genet. Mol. Biol.*, 3(1):16, 2004.

[38] M Baitaluk, X Qian, S Godbole, A Raval, A Ray, and A Gupta. PathSys: integrating molecular interaction graphs for systems biology. *BMC Bioinformatics*, 7(55), 2006.

[39] Z Hu, J Mellor, J Wu, and C DeLisi. VisANT: an online visualization and analysis tool for biological interaction data. *BMC Bioinformatics*, 5(17), 2004.

[40] GeneGo - System Biology for Drug Discovery. http://www.genego.com.

[41] A Nikitin, S Egorov, N Daraselia, and I Mazo. Pathway studio - the analysis and navigation of molecular networks. *Bioinformatics Applications note*, 19(0):1–3, 2003.

[42] M Hucka et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.