INTEGRATIVE ANALYSIS OF GENE NETWORKS USING DYNAMIC PROCESS PATTERN MODELLING

A. FREIER[+], M. LANGE[#], R. HOFESTAEDT
Bioinformatics Group, Faculty of Techology, Bielefeld University, Germany
afreier.or .hofestae@techfak.uni-bielefeld.de;
[#]*Plant Genome and Resource Center, IPK Gatersleben, Germany*
mlange@ipk-gatersleben.de
[+]*Corresponding author*

Keywords Network Modelling, Data Integration, Database Views, Object Databases

Abstract In this article a novel object-oriented modelling approach in the field of biochemical network modelling is presented. Molecular objects are modelled conceptually using object classes, internally based on the standard object models Java and ODMG. Objects and object networks are composed automatically using data integration. In combination with that, a specific view concept based on access paths has been implemented to model biochemical processes from integrated databases directly. Together with the application of graphical methods, networks are computed by the system. Each step of the workflow can be executed using a server and a graphical interface implemented with Java.

1. Introduction

Our goal is the computational construction and analysis of gene controlled metabolic networks (Russel, 1996). Biochemical processes involve a huge number of interconnected nodes, whereat efficient methods and tools are needed to support the access to different data sources, to integrate data retrieved from these sources and to apply common analysis methods, e.g. graph theory, mathematical simulation and visualization. This will give us an overview of topology and dynamics of cellular process networks and the occurrence of biochemical objects, e.g. the state of metabolic systems under the conditions of metabolic diseases.

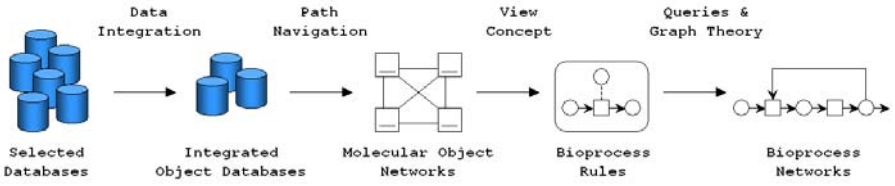

Figure 1. Process pattern modelling: mining biochemical networks from integrated data

Today, more and more internet databases are published providing selected molecular data concerning metabolism, gene networks and their application (Baxevanis, 2001). Actually, several systems for molecular database integration (Etzold et al., 1996; Stevens et al., 1999; Freier et al., 2002; Haas et al., 2001) are already used to get efficient access to distributed databases. At the same time information systems for modelling and visualization of regulative molecular networks are presented (Waugh, 2000;

Goesmann et al., 2002; Glass and Gierl, 2002). Still, even systems specialized at the same topic (e.g. gene networks) show differences in data modelling and in information content.

Actually, existing systems implement object services providing a previously defined object structure, where methods of application's exclusively are specialized to. Thus, a processing of user-specific objects is not possible. The main idea of the iUDB (Individually Integrated Molecular Databases) system is to provide a data independent toolbox for object database implementation, data integration, network modelling and analysis in application to genome data. Figure 1 shows the workflow accomplished by the system.

2. Methods and Algorithms

Three different models have been combined in our approach. In association with KDD (Knowledge Discovery in Databases) (Han, 1999) the initial step in our workflow starts with preprocessing the input data. To integrate data into our object database, we prefer the relational database model. iUDB has been enabled to access any data source, for which a JDBC (Java Database Connectivity) driver is available for. Like that, the data integration mechanism becomes independent from it's content.

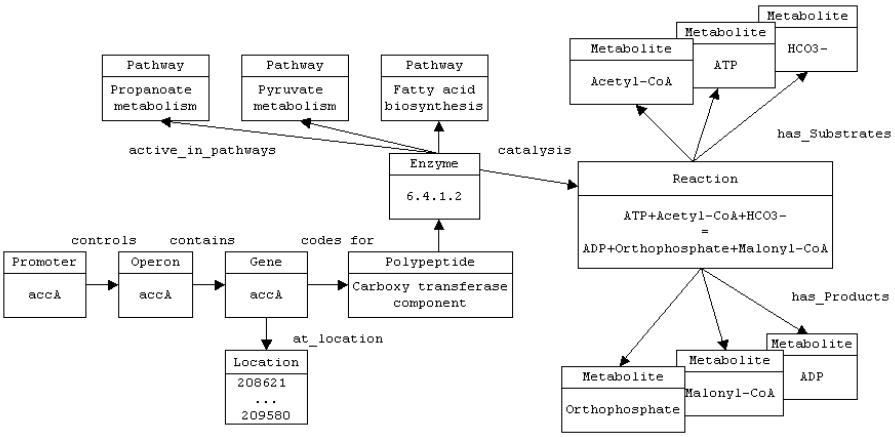

Figure 2. Example: molecular object network

For computational modelling of our biological data types we are using the object-oriented modelling paradigm. As we know, common methods (e.g. UML) and tools from software development are already available here. The user himself can select typical concepts including object classes, complex data types, inheritance and object references modelling the object type. Note that a network is described, if at least each object class refers to or is referred by other object classes. In the domain of gene networks, we are modelling, e.g. "Enzyme", "Pathway" and "Gene" classes. The object specification will be

internally mapped to an ODMG database (Cattel and Barry, 1997). However, by adding classes to the scheme, we include all structural information needed for the analysis of cell processes, which are bioprocesses (e.g. metabolic reactions), knowledge (relationships and logical conclusions between objects) as well as the physiology of biological structures (e.g. sequences, cells and compartments). An example of object networks is shown in figure 2. According to this example, the classes "Pathway", "Enzyme", "Reaction", "Metabolite", "Promoter", "Operon", "Gene", "Polypeptide" and "Location" have been modeled by the user. The interestingness of the approach is the fact, that all objects and their references, e.g. all things shown in the figure should be entered by the system automatically using, e.g. data integration. What we can do now is to navigate through access paths p=(c, {a_1, a_2, …, a_n}). Starting at "Promoter", we can use the path (Promoter, {controls, contains ,is_coding_for, is_active_in}) to retrieve all Pathways influenced by a given promoter.

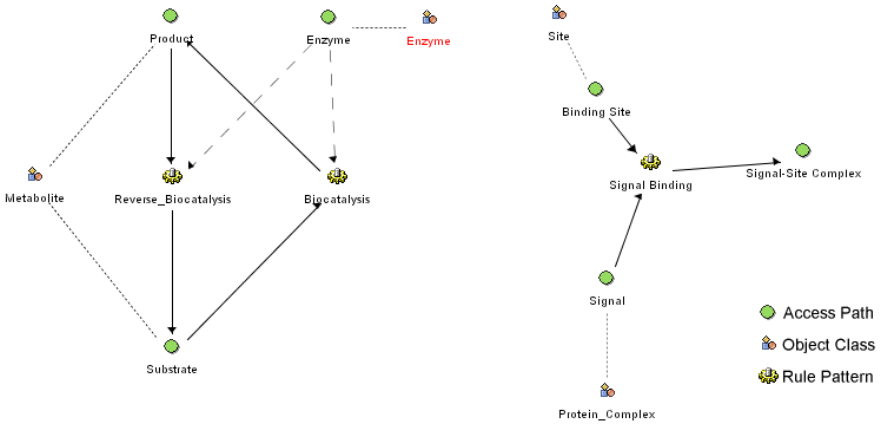

Figure 3. Bioprocess pattern examples

In order to model bioprocesses, a third model is used. In the center of the approach there is a rule based production system consisting of rules r = (S, P, I, L) used, where r itself represents the topology of a bioprocess, containing four elements. The sets S and P contain input and output objects, I is a set of influences and L specifies the location of the process described. To obtain these rules from a database a specific view concept has been developed. Starting at a defined root class, each rule pattern (or rule type) t access paths to each class, where S, P, I and L should be mined from. A model M=(T, R, N) holds the recent information, where T is a set of patterns, R is a set of rules mined by application of T against the database and N are the networks found by rule clustering (e.g. by application of graph-based searches). In figure 3 two examples are displayed. The left part of the figure shows two rule types modelling metabolic reactions. "Bio-Catalysis" rules transform "Substrate"

into “Product”, influenced by “Enzyme”, whereby “Substrate” and “Product” are different access paths to the “Metabolite” class and “Enzyme” is an access path to the “Enzyme” class. Note that the path specification is not displayed in the figure. The rule type “Reverse Bio-Catalysis” describes reversible reactions, meaning products are transformed into substrates. The next difference is selective. The user specifies class properties, in which dependency rules should be created or not. For the current example this property could be, e.g. a hypothetical class attribute “reversibility”, set to values similar to “reversible” or “not reversible”. In the second example in figure 3, a signal binding rules are modeled. Access paths lead to all object classes involved in the process (Protein Complex and Binding Site). Be stating that an aggregation of both classes is not part of the database scheme, the output of the process can not be specified directly. In short, the idea is to aggregate classes with each other. This means for our example, that the access path “Signal-Site Complex” is composed by the two access paths “Signal” and “Binding Site”. The application of both paths will be the elements of the aggregation. It is obvious that in our approach we finally combine and not merge different standard models for different applications.

3. Results and Discussion

The main result of our work is the implementation of the models and methods discussed in section 2 in the iUDB server program. In figure 4 an overview of the system is given.

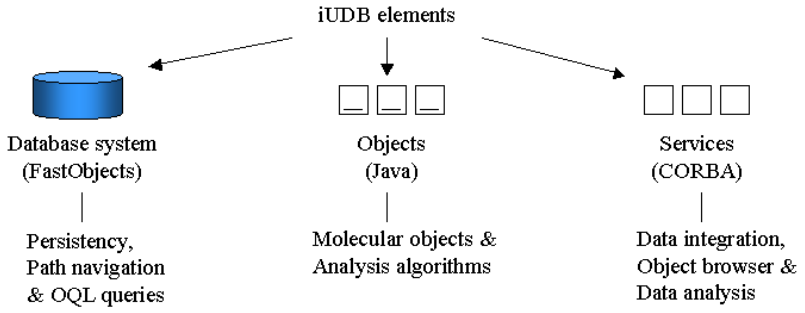

Figure 4. iUDB system elements and their implementation

All object classes and algorithms have been implemented in Java. Platform independent object access, analysis methods (e.g. advanced queries) and data integration capabilities are provided as CORBA services. Internally, we are using an ODMG conform object database (FastObjects), providing Persistency, Path navigation and OQL (Object Query Language) database queries. The graphical user interface of iUDB has been implemented in Java Swing. According to the tasks in figure 1 it contains modules for data source management, object integration, bioprocess views and network analysis.

3.1. Data Preprocessing

Data preprocessing includes the classification (Baxevanis, 2001) and management of all available data sources. iUDB contains a module for browsing data sources and data source management.

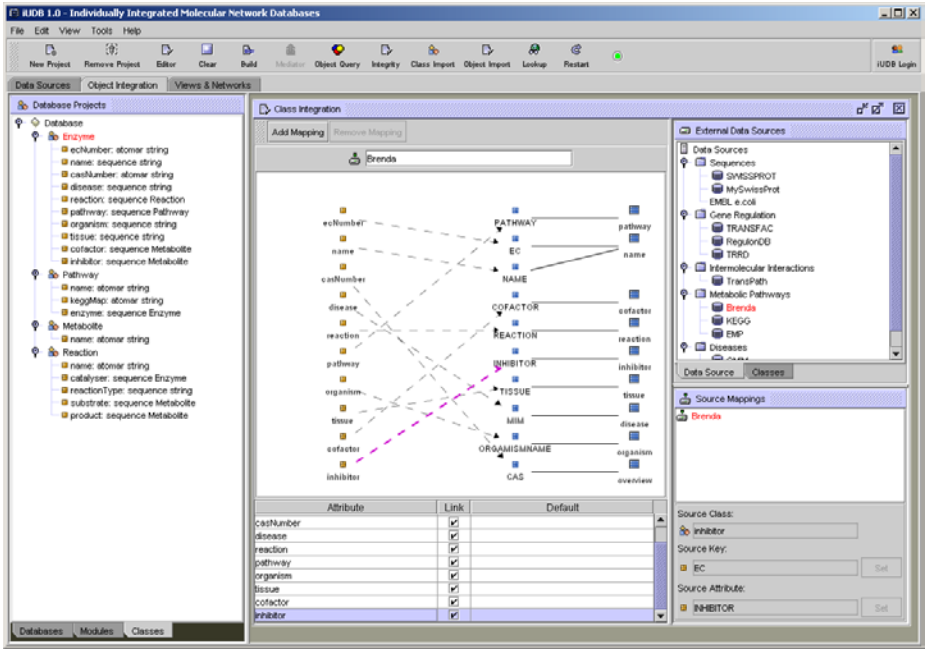

Figure 5. Data integration: retrieving complex objects from relational database sources

In figure 5 we can see the class editor, where networks of object classes can be modelled interactively. Also, there is an integration dialog, which enables the user to construct complex objects from a prepared list of data sources. For each object attribute mappings to different data sources, e.g. BRENDA (Schomburg et al., 1999), can be defined. Furthermore, objects can be reengineered from data source tables. However, we see the advantage of the object-oriented approach here: to obtain complex objects, complex join operation must be carried out by standard database systems. At the same time, the number of elements in the class diagram is much less than, e.g. in the same scheme modelled with relational databases. Summarized, the result of data preprocessing are integrated object databases with a user-defined data structure.

3.2. Bioprocess Views

As we have seen in section 2, the combination of object approach and rule system is useful to “animate” the object networks integrated in our database. The conceptual design of biochemical networks using our view concept dramatically reduces modelling time, compared to create processes or

transformation scripts, e.g. relational database views, by hand. The concept of access paths exhaustingly uses database references and could be implemented alternatively an enumeration of joins in SQL databases. Because of the lower complexity of operations and first measurements we expect a performance lead over standard SQL databases.

For efficiency, we snapshot rule views defined by the user by adding bioprocess rules to the database directly and interconnect them with the related objects.

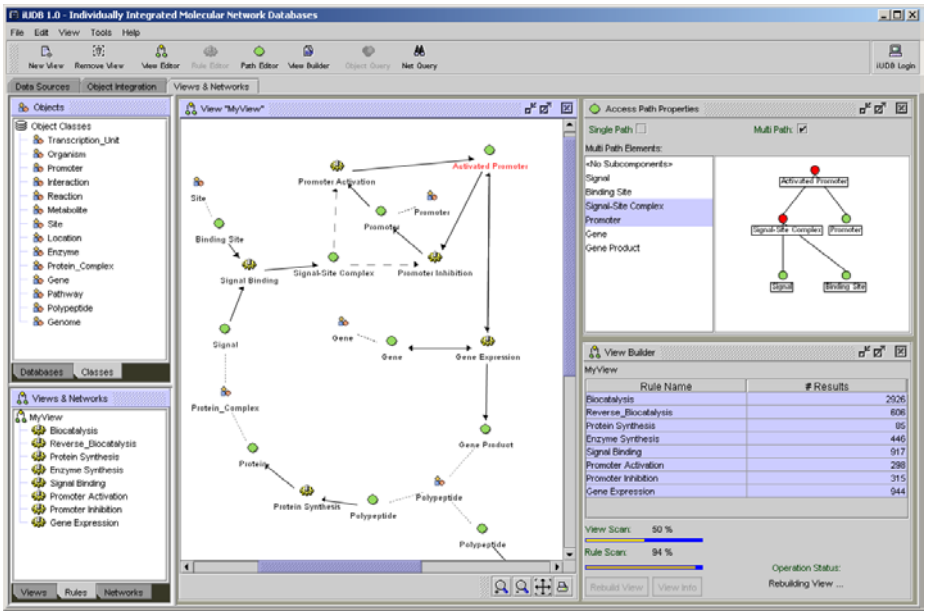

Figure 6. Modelling networks of bioprocess patterns using a specific view concept

So far, the resulting data is a bipartite graph consisting of objects of the object network, interconnected additionally by bioprocess rules. To improve query performance, we interlink objects and rules in a bi-directional way. In figure 6 we can see the view editor module enabling the user to design bioprocess views interactively.

3.3. Network Analysis

The analysis of integrated data using bioprocess patterns generates rules R for all patterns found in the database. They can be interconnected transitively as, e.g. pathways, by matching the identity (OID) of database objects they refer to. All rules of a network or pathway found are a subset of the rule set R. Networks can be stored in the system and loaded as a pre-processed network later. Actually, we implemented the following analysis methods:

- Transitive closure (object environment),

- Search for pathways between given objects,
- Computation of object interactivity profiles,
- Traversal of reaction paths and
- Typical set operations, e.g. union, difference and intersection

which are necessary for network comparison. Here the capabilities of the underlying ODMG compliant object database system is used.

Besides, the interactive pathway construction is possible and enhanced graphical methods will be implemented in the future development. Figure 7 shows the application of the system to the Glycolysis pathway. For each object in the database, the user can select consuming and producing rules to add them to the network.

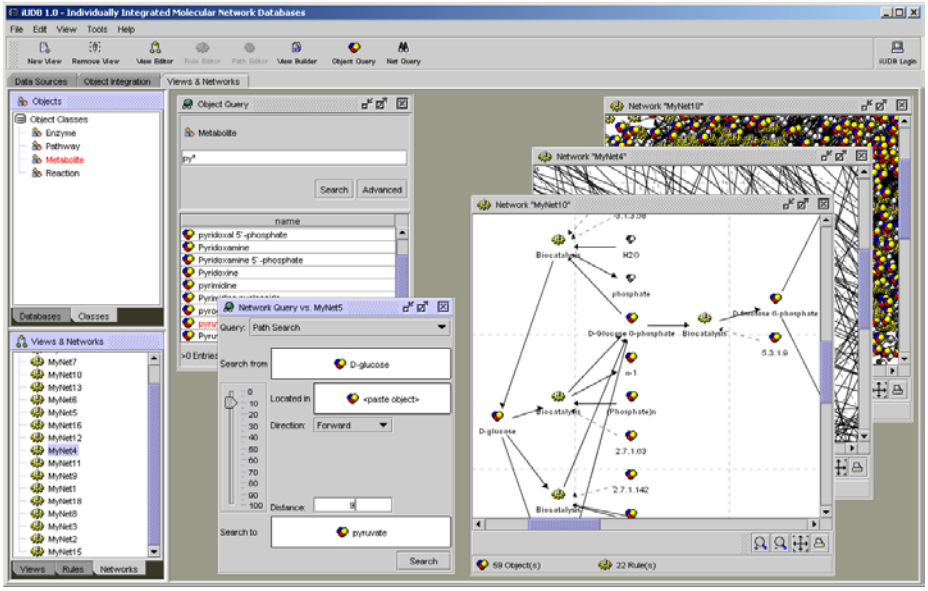

Figure 7. Applied graph theory: interactive analysis of biochemical objects and networks

3.3. Discussion

The idea of database integration applied to network analysis is not a new one. Moreover, it is a necessary task. Still, the growing number and the evolution of molecular databases demands adaptive systems able to integrate new data sources, build new integrated databases and apply suitable visualisation tools dynamically. With iUDB, we have developed a novel system supporting the object-oriented modelling and analysis of gene network and metabolic data. For data modelling, we used standard object models, which have been established in software development and already applied in bioinformatics.

Finally, the result is the interactive and automatic computation of bioprocess networks and pathways. In the near future, we will to continue the

workflow with specialised applications, e.g. graphical analysis, visualisation and simulation. The direct transformation of networks computed by iUDB into Petri-nets is an example. Actually, a first version of the system is available under the URL: http://tunicata.techfak.uni-bielefeld.de.

Acknowledgments
This work is supported by the German Research Council (DFG)

References

Baker, W., van den Broek, A., Camon, E., Hingamp, P., Sterk, P., Stoesser, G. and Tuli, M. A. The EMBL Nucleotide Sequence Database, Nucleic Acid Research 2000; 28, 19-23.
Baxevanis, A. D. The Molecular Biology Database Collection: an update compilation of biological database resources. Nucleic Acid Research 2001; 29, 1-10.
Cattell, R. and Barry, D. K. (eds) *The Object Database Standard: ODMG-93, Release 2.0.* Morgan Kauffmann Publishers, San Francisco, CA, 1997.
Etzold, T., Ulyanow, A. and Argos P. SRS: Information Retrieval System for Molecular Biology Data Banks. Methods in Enzymology 1996; 266, 114-128.
Freier, A., Hofestädt, R., Lange, M. and Scholz, U. BioDataServer: A SQL-based service for the online integration of life science data. In Silico Biology 2000; 2.
Glass, A and Gierl, L. A system architecture for genomic data analysis. In Silico Biology, Special Issue: GCB'01 2002.
Goesmann, A., Meyer, F., Kalinowski, J. and Giegerich, R. PathFinder: reconstruction and dynamic visualization of metabolic pathways. Bioinformatics 2002; 18, 124-129.
Haas, L. M., Schwarz, P. M., Kodali, P., Kotlar, E., Rice, J. E. and Swope, W. C. DiscoveryLink: A system for integrated access to life sciences data sources. IBM Systems Journal 40 2001; 489-511.
Han J. "Data Mining." In: J. Urban and P. Dasgupta (eds.), *Encyclopedia of distributed computing*, Kluwer Academic Publishers, 1999;
Kolchanov N. A., Ponomarenko M. P., Kel A. E., Kondrakhin Yu. V., Frolov A. S., Lopakov F. A., Kel O. V., Ananko E. A., Ignatieva E. V., Podkolodnaya O. A., Stepanenko I. L., Merkulova T. I., Babenko V. N., Vorobiev D. G. Lavryushev S. V., Ponomarenko Yu. V., Kochetov A. V., Kolesov G. B., Podkolodny N. L., Milanesi L., Wingender E., Heinemeyer T. and Solovyev V. V. GeneExpress: a computer system for description, analysis and recognition of regulatory sequenecs of the eukaryotic genome. ISMB, 6:95-104. MEDLINE PMID: 9783214; UI: 98456543, 1998.
OMG The Common Object Request Broker Architecture: 2.0/IIOP Specification, OMG Document Number 96.08.04. OMG (Object Management Group) 1996.
Russel, P. J. *Genetics, Fourth Edition.* Harper Collins Publishers, NY 1996.
Schomburg, D., Schomburg, L., Chang, A. and Bänsch, C. BRENDA the Information System for Enzymes and metabolic Information, In: Proceedings of the German Conference on Bioinformatics (GCB '99), Germany, pp. 226-227, 1999.
Stevens, R., Baker, P. and Bechofer, S. TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. Bioinformatics 2000; 16, 184-185.
Waugh, M. Pathdb helps researchers analyze metabolism. Technical Report 1, National Center for Genome Resources, 2000.