SIGBioinformatics Record

Newsletter of the SIGBioinformatics
ACM Special Interest Group

Volume 2, Issue 3, September 2012 ISSN 2159-1210

SIGBioinformatics officers, Committees, Awardees

SIGBioinformatics Officers

Chair
Aidong Zhang

Vice-chair
Gultekin Ozsoyoglu

Secretary-Treasurer
Armin R. Mikler

Newsletter Editor
Pietro Hiram Guzzi
Young-Rae Cho

Board of Directors

Vasant Honavar
Sun Kim
Vipin Kumar
Yi Pan
Marie-France Sagot
Cathy Wu

Director of Conferences

Vasant Honavar

Awards Committee

Umit V. Catalyurek (Co-Chair)
Wei Wang (Co-Chair)
Constantin Aliferis, M.D., Ph.D.
Jonathan Dennis
Tony Hu
Parthasarathy Srinivasan
Joe Zhang
Zhongming Zhao

Information Director

Mohammed Zaki

Associate Information Directors

Guo-zheng Li
Li Liao
Yunlong Liu

Liaison to Industry and Other Societies

Sanjay Ranka

Advisory Board

Mark Borodovsky (Chair)
Pierre Baldi
Jeff Bennetzen
Volker Brendel
Robert Cottingham
Dmitrij Frishman
Mark Gerstein
Nick V. Grishin
Minoru Kanehisa
Nikos Kyrpides
Meral Ozsoyoglu
Sir Richard John Roberts
Dong Xu

Table of Contents

1. Editor-in-Chief's Notes

With this issue, we are starting to change the SIGBioinformatics Record in order to provide, hopefully, a more useful and interesting newsletter to our SIGBioinformatics community. This issue contains three new sections, as listed below.

- *Editor's Notes*, that provides an overview of the structure and contents of the newsletter,
- *Contributed Articles*, edited from now on by the Contributed Articles area editor, Young-Rae Cho. Contributed articles are to be short papers or reviews (even discussing ongoing projects or works). The area editor is responsible for soliciting authors, collecting papers, and having the submitted papers reviewed.
- *Systems and Prototypes*, edited in this issue by me, and, in the future issues, by a soon-to-be-appointed area editor. This section includes solicited/unsolicited short bioinformatics software/system descriptions. The area editor is responsible for soliciting authors, collecting papers, and having the submitted papers reviewed.

And, in the next issue, we plan to open up a new section, namely, "Research Centers", that will be reporting about the leading bioinformatics computer centers in the world. The research centers area will also have a new area editor, responsible from the articles in the area.

Enjoy the issue!

Pietro Hiram Guzzi,

Newsletter Editor-in-Chief

2. Contributed Articles

Editorial: Contributed Articles on SIGBioinformatics Records

The last decade has witnessed revolutionary breakthroughs in biotechnology, accompanied with the rapid accumulation of genome-wide biological data as well as various types of biomedical or clinical data. At the same time, bioinformaticians have faced a great challenge; how to manage and mine such large-scale, complex data for significant knowledge discovery. Recent research in bioinformatics has highlighted the application of effective computational approaches to tackle this challenge.

The section "Contributed Articles" on SIGBioinformatics Records presents high-quality research introducing new development of cutting-edge technology and demonstrating biologically influential results. The main purpose of this section is to address open issues in all computational aspects of bioinformatics and biomedicine including algorithms, modeling, simulation, databases and applications. In this issue, two recent research articles are presented.

A data warehouse is semantically consistent data storage, constructed by integrating data from multiple heterogeneous sources, to support structured and analytical queries using OLAP operations. Because data warehouses are specifically designed based on a multidimensional data model, e.g. the star schema model, they have been widely used for analysis of biomedical data. Due to diversity and complexity, biomedical data are typically organized in a form of multiple dimensions. Du et al. in "BioStar+: A Data Warehouse Schema for Integrating Clinical and Genomic data from HIV Patients" presents a novel data warehouse schema, called BioStar+, to effectively manage HIV patient data including clinical data, gene data, experimental data and microarray data. This model is an upgraded version of the BioStar schema proposed previously by the authors. The BioStar model has significant advantages in applicability and extensibility. However, as a downside, it has restriction on usage of data spaces during querying. The newly proposed BioStar+ model overcomes this limitation of BioStar by maintaining the original data spaces same as the star schema.

Gene Ontology (GO) is a repository of biological ontologies and annotations of genes or proteins. Although the annotation data on GO have been created by the published evidence resulting from mostly unreliable high-throughput experiments, they are frequently used as a benchmark for functional characterization because of their comprehensive information on the genomic scale. A typical example of the application of GO is to quantify functional similarity between proteins by measuring semantic similarity. Because semantic similarity is a function that returns a numerical value reflecting closeness in meaning between GO terms, it is able to estimate functional similarity between two proteins annotated to any GO terms. Guzzi and Mina in "Towards the assessment of semantic similarity analysis of protein data: approaches and issues" introduces semantic similarity measures and the issues in use of current GO and its annotation data.

I would like to thank all the authors for their high-quality work contributed to this section.

Young-Rae Cho, *Editor*
Department of Computer Science
Baylor University

BioStar+: A Data Warehouse Schema for Integrating Clinical and Genomic data From HIV Patients

Nan Du, Suxin Guo, Supriya D Mahajan, Stanley A. Schwartz,

Bindukumar B Nair, Chiu Bin Hsiao, Aidong Zhang

State University of New York at Buffalo

{nandu, suxinguo, smahajan, sasimmun, bnair, chsiao, azhang} @buffalo.edu

ABSTRACT

In the field of biomedicine, it is becoming increasingly apparent that a huge amount of genomic and clinical data are being generated everyday. Although these data, stored in different formats at multiple sources, help us acquire more information about the patients, the scale and complexity of these datasets result in the challenges of integration of multiple biomedical datasets. Based on the traditional star schema and our previous Biostar schema we propose a hybrid schema called BioStar+ for HIV patient data modeling. By maintaining the original data space structure instead of disassembling it, this hybrid schema not only retains the original advantages from BioStar, but also improves the query efficiency. In addition, we describe the HIV patients data warehouse that we have developed based on the BioStar+ schema and discuss an analysis case based on this schema.

Keywords

BioStar+, Multidimensional Modeling, Data Warehouse Schema

1. INTRODUCTION

Large amounts of data in the field of biomedicine research, ranging from clinical data to gene expression data, are being generated. Thanks to these data, we may deeply understand the biological mechanisms behind diseases. For example, clinical data which refers to any information contained in a patient's medical record helps the researchers accurately and appropriately trace the performance of various kinds of medicines and microarray data contain valuable information for discovery of disease-associated gene expression patterns and classification of patients. However, the scale and complexity of these datasets also make the database research in biomedicine a big challenge. This is because of the rich variety of data which are easily available from genome sequences and protein structural data of organisms, and the large quantities of data that are becoming available through modern experimental techniques. This raises the issue of integrating and analyzing these data which can lead to a better understanding of biological functions at all levels [1, 6]. If we want to obtain full benefit of functional genomics, we need to find a way of seamless integration with large amounts of patient datasets in the field of biomedicine.

Data warehouse has been widely used to support analysis of medical data [7, 10]. Data warehouse is generally used to provide analytical results from multidimensional data through effective

summarization and processing of segments of source data relevant to the specific analyzes. Data warehouse can integrate heterogeneous data from multiple and distributed information sources, which is considered as the basis of DSS (Decision Support Systems) that provides analytical results to managers or researchers so that they can analyze a situation or a disease and make important decisions [5].

Schema design for a biological data warehouse is usually based on the notion of the star schema. It is called a star schema because the entity-relationship (ER) diagram of this schema resembles a star, whose links radiating from a central fact table. Furthermore, star schema is composed of one or more fact tables which contain the primary information and numbers of dimension tables, each of which contains information about the entries for a particular attribute in the fact table. Snowflake is a special case of star schema whose foreign keys can be embedded in the dimension table so that a certain dimension table could have relationships with other dimension tables. Similar to star schema, it is so called a snowflake schema because the diagram of the schema resembles a snow flake. Use of the star or snowflake schema aimed at limiting accesses and queries in a data warehouse environment. Although both of star schema and snowflake schema have a relatively simple structure, they are widely used in practice. A multidimensional data model for clinical data that extended the star schema was proposed by Pedersen et al [9]. Object-oriented and traditional entity relationship (ER) approaches are also used to model the biological data. Paton et al. [8] proposed an object-oriented conceptual model for various yeast data, including genetic, genomic sequence, gene expression and protein-protein interaction data. These models were also implemented in an object database called GIMS by Cornell et al [3]. In addition, many others used traditional ER models for microarray gene expression or other genomic data [2, 4].

In this paper, we present a data warehouse schema named BioStar+ for integrating clinical and genomic data collected from HIV affected patients. BioStar+ is a hybrid schema which combines the star schema with our previous BioStar schema [7]. This new schema has greater query efficiency and is easier for users to understand than the existing methods.

The rest of the paper is organized as follows. In Section 2, we discuss some motivations for the new biomedicine data warehouse modeling. We then describe a new model called BioStar+ in Section 3. In Section 4 we describe an application process for BioStar+. Finally, we conclude with a brief discussion and summary in Section 5.

2. MOTIVATION

Even after 25 years of the AIDS epidemic, AIDS remains the deadliest epidemic in human history, killing more than 25 million people worldwide, including more than 500,000 Americans. Fortunately, major developments in the field of molecular biology, coupled with strides in genomic technologies, have led to an explosive growth in HIV patient data, including clinical data, gene data, experiment data and microarray data, etc. With these data, we may obtain valuable new insights about AIDS. Due to the diversity and complexity of these data, we need to integrate relevant information from these vast datasets with an effective data warehouse schema. Among these various datasets, we model the data warehouse schema by using the following four data spaces: microarray data space, gene data space, experiment data space and clinical data which will be discussed detailedly in Section 3.

We now review the structure of BioStar. Figure 1 shows the typical structure of BioStar schema, which has one table as the fact table (Fact), four m-tables (MTable1-4) and four dimension tables

(Dim1-4). In a BioStar model, an m table includes the primary key of the central entity table and the primary keys of the associated dimension tables as the foreign keys of the m-table may be associated with multiple dimensions. The m-tables often represent many-to-many relationships between the fact table and dimension tables. An m-table may contain 0, 1, or numbers of measures. Each m-table may also include non-measure attributes that are used to characterize the relationship between the central entity and dimensions, or specify the temporal validity of the measure. The dimension tables in a BioStar schema may be normalized as in a star schema, or normalized so that explicit dimension hierarchies can be defined [11].

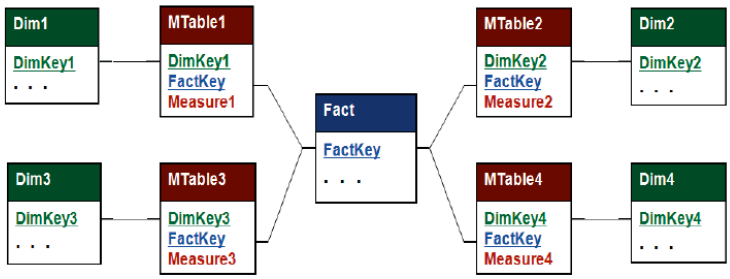

Figure 1: Typical structure of a Biostar schema.

As analyzed in [11], there are several advantages of modeling with the BioStar schema. First, the BioStar model has the property of great extensibility by storing different measures in separate m-tables. In such a configuration, an existing dimension table and its associated m-table can be modified independently from the other dimensions and mtables. When a new dimension table needs to be added to the warehouse, we just need to create a new dimension table and a new m-table without affecting the existing tables. The new m-table simply uses the primary key of the fact table as one of the foreign keys, then establish the relationship between the new dimension table and fact table. Second, the many-to-many relationships between the fact table and dimension tables could be handled by the m-tables. We just need to create a m-table which is used to connect the fact table and dimension table. Third, uncertain relationships between the fact table and dimension tables may be kept in the m-tables. An additional field may be retained in the mtable to specify if a relationship instance is uncertain. Furthermore, an m-table is much smaller than the fact table of a star schema. Fourth, the BioStar model allows an m-table to have non-measure attributes. Finally, the BioStar model can be used to handle the incomplete data from biomedicine studies. If incomplete data are stored in the fact table of a star schema, null values may need to be used for some missing measures and their associated dimension keys [11].

However, Biostar schema also has some restrictions on cross data spaces analyzing or querying. We now discuss the situation when we want to make an analysis through several data spaces, in other words, we need to integrate several data spaces together. In this case, we need to consider all the tables from these data spaces as dimension tables and link to the central fact table. For example, if we have a BioStar schema HIV data warehouse which integrates the clinical data space, gene data space and experiment data space together, then we need to integrate all tables in three spaces together as dimension tables. Since there would be tens of even hundreds of tables connect to the fact table directly as dimension tables or with m-tables, this may result in two problems. On the one hand, it may reduce the effectiveness of browsing or querying. As we known, most of the queries are limited only in a certain data space. In such a configuration, we have to connect to the huge fact table even we just want to query some information from a certain data space. For example, if we want to

query how many kinds of drug a patient has taken in the past two years, this question can be answered just by using the drug table and patient table in a clinical data space. But now, most queries have to connect to the huge central fact table, because only by this way different dimension tables could be connected to each other. On the other hand, if all the tables in different data spaces are connected to the fact table directly, this structure would be hard to understand, which may lose the original biological meanings.

3. BIOSTAR+ SCHEMA

In this section, we describe a new multidimensional model for HIV patient research, called BioStar+, which is a hybrid schema based on Biostar schema and star schema. As we mentioned before, the complexity of the data involved in HIV research suggests using four modeling data spaces: clinical data, gene data, experiment data, and microarray data. These conceptual data spaces will be used to describe our BioStar+ schema, a part of which have been cited from [11]. First of all, let us discuss the main modeling and characteristics of these data spaces.

3.1. Data Spaces for HIV Patient Research

Each data space is shown by an ER (entity-relationship) diagram, where each rectangle denotes an entity with the entity name. Furthermore, a relationship between two entities is shown by a line with the multiplicity label, which indicates the number of objects that may participate in the relationship. There are mainly three relationships between two entities, which are “one-to-one” denoted as “1-1”, “one-to- many” demoted as “1-n” and “many-to-many” demoted as “n-n”.

3.1.1. Clinical data space

An ER diagram of the clinical data space is shown in Figure 2. The clinical data space has a rich variety of entities, among which Patient is obviously the most important entity. Disease, Drug and Clinical Test have a many-to-many relationship with Patient. Clinical Sample is another important entity, which has a many-to-one relationship with Patient. Clinical Samples such as blood samples are taken from patients and used for various laboratory assays. Demographics, which characterizes patients based on demographic information has a many-to-one relationship with Patient. Follow up, which is used to capture patient status information, also has a many-to-one relationship with Patient. The Clinical Test entity captures information about simple clinical tests applied directly to patients through physical examination or by asking patients to perform some simple routines. Furthermore, Clinical Test has a many-to-many relationship with the Patient entity. The real clinical data space may have a very complex structure and may include more entities such as patient medicine history record or patient family allergy history. Different dimensions characterize patients with different fact measures. However, for a particular patient, we just consider the measures that are usually available.

The star schema of the clinical data space is shown in Figure 3. Patient is the fact table, which has PatientID, an unified patient identifier as the primary key. The dimension table FollowUp, ClinicalSample and Demographic, which have a one-to-many relationship with the fact table, embed their primary keys in the fact table so that they can have direct relationship with the fact table. For the case of the many-to-many relationship between dimension table and fact table, such as Drug, Disease and ClinicalTest, an m-table is created, which is similar to the treatment of many-to- many relationships in a standard ER schema.

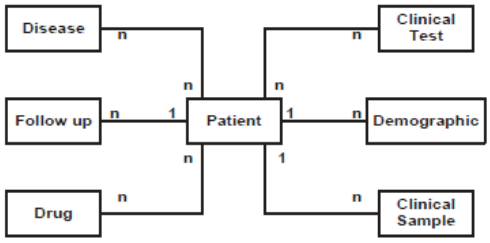

Figure 2: A ER diagram for the clinical data space.

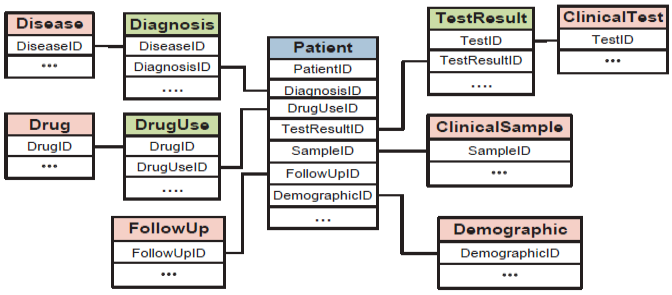

Figure 3: A star schema for the clinical data space.

3.1.2. Gene data space

An ER diagram for the gene data space which contains gene function information integrated from a variety of public domain data sources is shown in Figure 4. Gene clusters are obtained by clustering analysis of gene expression data in the warehouse and can be used to analyze gene regulatory networks. Promoter describes the information between gene expression data and promoters. Protein Expression describes another level of gene expression measurement using proteomics approaches and has a many-to-one relationship with Gene Sequence. Protein Domain information is also an important dimension table for describing gene functions. The other gene data includes Protein Expression, which represents another level of gene expression measurement and has a many-to-one relationship with Gene Sequence.

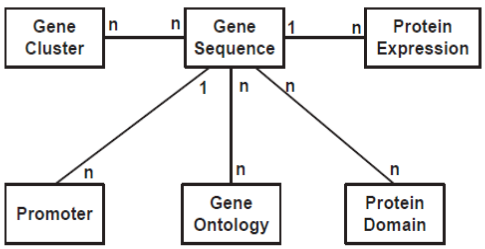

Figure 4: A ER diagram for the gene data space.

The star schema of the gene data space is shown in Figure 5. GeneSequence is the fact table, which has GeneSequenceID, an unified patient identifier as the primary key. The dimension table Promoter and ProteinExpress, which have a one-to-many relationship with the fact table, embed their primary keys in the fact table so that they can have direct relationship with the fact table. For the case of the many-to-many relationship between dimension table and fact table, such as GOTerm, Cluster and DomainModel, an m-table is created for them to connect to the fact table.

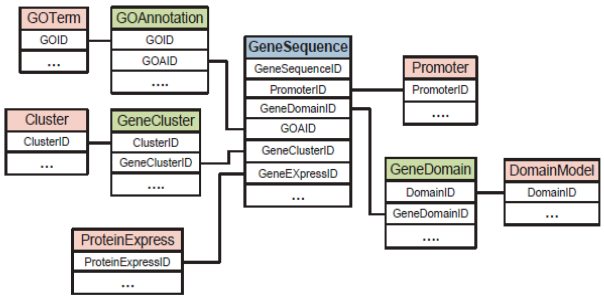

Figure 5: A star schema for the gene data space.

3.1.3. Experiment data space

An ER diagram for the experiment data space which contains a variety of experiments information is shown in Figure 6. All of Project, Publication, Protocol, Platform and Normalization have a many-to-one relationship with Experiment.

The star schema of the gene data space is shown in Figure 7. Experiment is the fact table, which has ExperimentID, an unified their identifier as the primary key. All the tables have a one-to-many relationship with Experiment, thus they embed their primary keys in the fact table so that they can have direct relationship with the fact table.

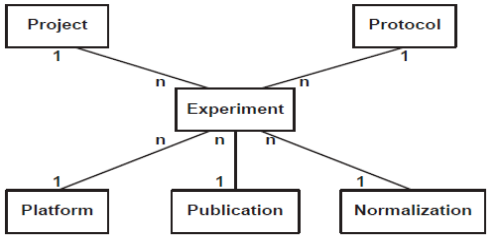

Figure 6: A ER diagram for the experiment data space.

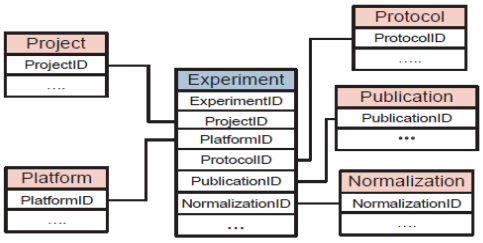

Figure 7: A star schema for the experiment data space.

3.1.4. Microarray data space

The gene data space is about the gene function information integrated from a variety of domain data sources. The fact object in the gene data space is the mRNA Expression. The mRNA Expression has a direct relationship with clinical and experiment, which are the fact tables in the relative data space. In addition, mRNA Expression has an indirect relationship with gene, the fact table in the gene data space, through an array probe entity. Array probe captures the information about sequence or probes that are placed on the microarray. Since multiple probes, which are derived from gene

sequences may be used for a single gene, it is often necessary to summarize gene expression to the higher level of non-redundant gene sequences.

Taking the Affymetrix GeneChip platform as an example, it provides two kinds of gene expression measurements for each probe set, a presence/absence (PA) call and a numeric value. Obviously, mRNA Expression can be considered as the center of these four data spaces, because it has a direct or indirect relationship to other data spaces. Besides that, a measurement unit entity, which is used to keep the information about what is measured for gene expression, has a many-to-one relationship with mRNA Expression. An ER diagram for the microarray data space which connects to the other data spaces is shown in Figure 8. The ER diagram of microarray data space would not be shown because we will use this data space as the fact table of our BioStar+ schema.

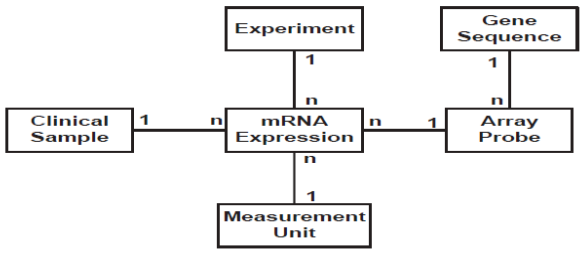

Figure 8: A ER diagram for the microarray data space.

3.2. The description of the BioStar+ schema

The schema of the BioStar+ can be defined as W = (F, S, D, N, M, C), where F is the fact table, S is a set of space-fact tables, D is a set of dimension tables, N is a set of space-measure tables, M is a set of measure tables and C is a set of summarisability constraints. The fact table is the core of the BioStar+ schema, which is associated with each space-fact table. The fact table defined in BioStar+ is di□erent from the fact table defined in star schema or snowflake schema. The most obvious di□erence between them is the fact table defined in BioStar+ is associated with space-fact tables instead of dimension tables in star schema or snowflake schema. S, the set of space-fact tables, plays a similar role as the fact table defined in star schema or snowflake schema. Both of them are associated with the dimension tables. The only di□erence is that, in the BioStar+, we define that all the dimension tables that connect to the same space-fact table are from the same data space. D, the dimension tables, stores the attributes in a certain data space. Furthermore, all of F , $S_i \in S$ and $D_i \in D$ have a pair of (L, □) structure, where L is a set of dimension levels which is associated with a domain of values and □ is a partial order of the elements in L. However, since our discussion mainly focuses on modeling the data warehouse schema, the discussion of classification hierarchy would not go into details.

N, the space-measure tables, are associated with the fact table and one or more space fact tables. Note that, a space- measure table is used only when there is a many-to-many relationship between the fact table and a space-fact table. Similarly, a measure table M, which is used only when there is a many-to-many relationship between a space-fact table and a dimension table, is associated with the space-fact tables and one or more dimension tables. Both $N_i \in N$ and $M_i \in M$ have a triple (A_m , A_s , T_m) structure, where A_m is a set of attributes called measures, A_s is a set of supporting attributes for the measures and T_m is a set of dimensions where $T_i \in D$ or S. Note that Am can be an empty set.

Finally, each constraint, $C_i \in C$, is a triple (E_i , R_j , α), where $E_i \in D$ or S, $R_i \in N$ or M , and α is an aggregation operator.

3.3. BioStar+ schema on the sample spaces

Figure 9 shows the typical structure of the BioStar+ schema based on the clinical data space, gene data space, experiment data space and microarray data space, which has one fact table (Microarray Expression), three space-fact tables (GeneSequence, Clinical, Experiment and measurement unit), one space-m table (ArrayProbe), sixteen dimension tables (GOTerm, Cluster, Promoter, DomainModel, Disease, Drug, ClinicalTest, ClinicalSample, Project, Platform, Protocol, Publication, Normalization, ProteinExpress, FollowUp and Demographic) and six m-tables (GOAnnotation, GeneCluster, GeneDomain, Diagnosis, DrugUse and TestResult). Space-m table is used to connect the fact-space table to the fact table, when there is a many-to-many relationship between them. For example, as Figure 9 shows, GeneSequence has a many-to-many relationship with microarray expression, thus they could not connect to each other directly. With a space-m table ArrayProbe, we can transfer this "many-to-many" relationship to two "many-to-one" relationships.

Each space-fact table is the core of a data space which works like a fact table in the star schema. Note that each space-fact table which blocks the dimension tables connecting to the fact table (Microarray Expression) directly is playing a fact table's role in a certain data space. Note that, measurement unit is a special case which does not have dimension tables. Although measurement unit is not a core of a data space, it has an close relationship directly with microarray expression and would be used frequently together with microarray expression, thus we consider it as a space-fact table. Furthermore, each space-fact table links to the fact table with the same primary key MicroExpressID which works like a fact key in the BioStar schema. Note that, each space-fact table and the associated m-tables and dimension tables describe a data space independently, and the schema of them is a star schema. Therefore, it is the reason why we say our BioStar+ schema is a hybrid schema which represents the relationships between the space-fact tables and the fact table with a BioStar schema and the relationships among space-fact tables, m-tables and dimension tables with a star schema. Each m-table represents a many-to-many relationship between the space-fact tables and dimension tables.

Each m-table includes one of the foreign keys of the space-fact table and the primary key of the associated dimension tables. An m-table may contain 0, 1, or numbers of measures. When an m-table includes non-measure attributes, it is used to characterize the relationship between the space-fact tables and dimension tables, or to specify the temporal validity of the measure. Finally, the dimension tables are one of the set of companion tables to a space-fact table, which are kept on each dimension that decision makers would like to either roll up or drill down.

The BioStar+ schema has maintained all the advantages described from BioStar in Section 2, since we has maintained the main structure of the BioStar schema. Besides, the BioStar+ schema also has several extra advantages. First, the BioStar+ schema has the property of improving the effectiveness of browsing and querying. As we know, most of the queries are limited only in a certain data space. By the BioStar+ schema, the tables from a certain space are now modeled in a star schema which maintains the original space relationships among ER tables.

Modeling with a BioStar+ schema, we do not need to connect to a huge fact table whenever we just want to browse or query something in a certain data space. For example, in the schema shown in Figure 9, if we want to query how many papers have been published under a certain project, we just

need to join the Project table and Publication table through the space-fact table Experiment without linking to the fact table Patient. In addition, BioStar+ is easily extensible by adding new attributes to the space-fact table or to one or more of the dimension tables and new dimension tables to the schema without interfering with the fact table. Finally, due to the property of star schema which is represented by centralized space-fact tables which are connected to multiple dimensions, the BioStar+ is easier to understand. As shown in Figure 9, each data space is described relatively independent with a star schema, whose center is the space-fact table. These space-fact tables (Clinical, GeneSequence, Experiment and measurement unit) link to the fact table with the unified identifier key MicroExpressID directly or indirectly, which is shown as dash lines in Figure 9.

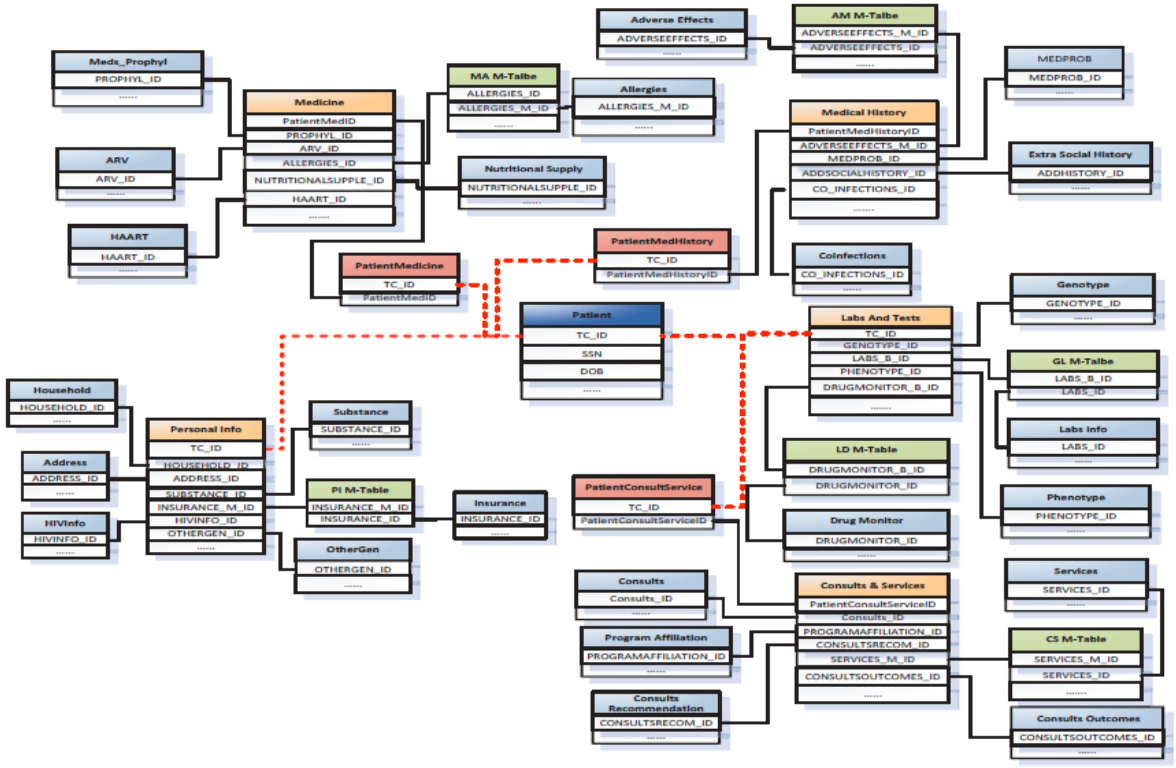

Figure 9: BioStar+ schema on the HIV data spaces.

4. APPLICATION

In this section, we describe the application of the BioStar+ schema to the development of our HIV patient data warehouse, which integrates experiment space data, gene space data, clinical space data, and provides support for effective data mining and querying. Let us see how this data warehouse can be used for defining more complex gene expression exploration operations. Here, we give a case study for using our HIV patient data warehouse to integrate disease information with microarray gene expression data. We will take our HIV dataset as an example. Our clinical database is derived from the patients enrolled in the Immunodeficiency Services Clinic of the Erie County Medical Center, Buffalo, NY. Our detailed gene data come from cDNA microarrays with RNA obtained from HIV patients.

This analysis may involve three data space: clinical data space, gene data space and measurement data space. The clinical dimension captures the information about the patients with medicine and patients without medicine. As mentioned before, in measurement data space, the Affymetrix

GeneChip platform provides two kinds of gene expression measurements: a presence/absence (PA) call and a numeric value. The PA call can take one of the following three labels: “P” for “present”, “M” for “marginal” and “A” for “absent” of gene expression.

For example, one of our goals is to identify informative genes that are unique to the NP and LTNP patients by using data mining operations, where NP denotes normal progressors who develop AIDS 3-4 years post infection and LTNP denotes the long term non-progressors who do not develop AIDS for > 15 years post infection. The process of identifying informative genes for NP and LNTP has four steps. First, we summarize the gene expression data by using measurement unit. If the PA call is “A”, we set the gene expression value to 0, otherwise leave the value unchanged. Second, we set the gene expression data over the clinical dimension, which is “normal progressors” for NP and “long term non-progressors” for LNTP. Third, the slice of NP and LNTP are selected. Then T test is applied to calculate a p-value for each gene. Finally, over the gene data space, we filter the genes with a p-value threshold of 0.05. A gene with a p-value less than 0.05 is considered as an informative gene whose expression is changed in NP when compared with LNTP.

Access to our HIV data warehouse can be obtained by browsing through the database or by using the query interface. A website which is used for querying and browsing the HIV data warehouse runs on the Oracle 10g database. One of the query interfaces of the HIV data warehouse is shown in Figure 10. This figure shows a query interface of clinical data space which includes the query options of personal information, medicine history, allergy history, follow up information, etc, from which the users could find out any clinical information about an HIV patient.

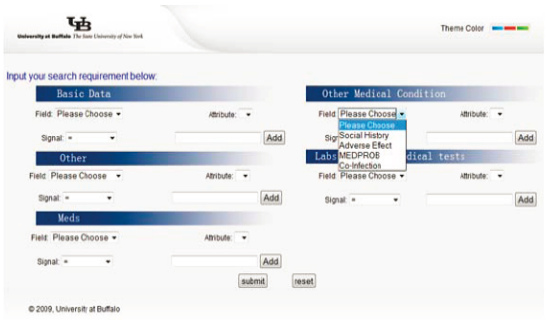

Figure 10: Query interface of our HIV data warehouse.

5. CONCLUSIONS

The major challenge in the integration of HIV patient data is the large number of distributed, semantically disparate data sources that need to be integrated into an effective data warehouse. In this paper, we presented a data warehouse schema named BioStar+ for managing and exploring HIV patient data. BioStar+ is a hybrid schema based on our previous BioStar and star schema which uses a star schema to model each data space and integrates them with a BioStar schema. Besides retaining the advantages of the original BioStar, BioStar+ also have the characteristics of high query efficiency and understandability. Moreover, we showed that this data warehouse schema can be used for defining more complex gene expression exploration operations based on an HIV data warehouse we have recently developed.

6. REFERENCES

[1] A. Bayat. Science, medicine, and the future: Bioinformatics. *British Medical Journal* (BMJ), 324:1018–1022, 2002.

[2] J. Chen, P. Zhao, D. Massaro, L. B. Clerch, R. R. Almon, D. C. DuBois, W. J. Jusko, and E. P. Hoffman. The PERP GeneChip data warehouse, and implementation of a dynamic time series query tool (SGQT) with graphical interface. *Nucleic Acids Research*, 32(Database issue):D578–D581, 2004.

[3] M.Cornell, N. Paton, S. Wu, C. Goble, C. Miller, and P. Kirby. GIMS - a data warehouse for storage and analysis of genome sequence and functional data. *Proceedings of the 2nd IEEE international symposium on bioinformatics and bioengineering*, pages D578–D581, 2001.

[4] J. Gollub, C. A. Ball, G. Binkley, J. Demeter, D. B. Finkelstein, J. M. Hebert, T. Hernandez-Boussard, H. Jin, M. Kaloper, J. C. Matese, *et al.* The Stanford Microarray Database: data access and quality assessment tools. *Nucleic Acids Research*, 31(1):94–96, 2003.

[5] L. Y. K. Judice. Correlation-based methods for biological data cleaning. PhD thesis, 2007.

[6] N. Khan. A cooperative framework for molecular biology database integration using image object selection. PhD thesis, 2004.

[7] M. Levene and G. Loizou. What works. Data warehouse: decision support solution reduces patient admissions, saves payer millions. *Health Management Technology*, 20:42–46, 1999.

[8] N. W. Paton, S. A. Khan, A. Hayes, F. Moussouni, A. Brass, K. Eilbeck, C. A. Goble, S. J. Hubbard, and S. G. Oliver. Conceptual modelling of genomic information. *Bioinformatics*, 16(6):548–557, 2000.

[9] T. B. Pedersen, C. S. Jensen, and C. E. Dyreson. A foundation for capturing and querying complex multidimensional data. *Information Systems Journal*, 26(5):383–423, 2001.

[10] R. Scheese. Data warehousing as a healthcare business solution. *Healthcare Financial Management Journal of the Healthcare Financial Management Association*, 52(2):56–59, 1998.

[11] L. Wang, A. Zhang, and M. Ramanathan. Biostar models of clinical and genomic data for biomedical data warehouse design. *International Journal of Bioinformatics Research and Applications*, 1(1):63–80, 2005.

Towards the Assessment of Semantic Similarity Analysis of Protein Data: Main Approaches and Issues

Pietro Hiram Guzzi
University of Catanzaro

Marco Mina
University of Padova

ABSTRACT

Bioinformatics approaches to the study of proteins yield to the introduction of different methodologies and related tools for the analysis of different types of data related to proteins, ranging from primary, secondary and tertiary structures to interaction data [1], not to mention functional knowledge.

One of the most advanced tools for encoding and representing functional knowledge in a formal way is the Gene Ontology (GO) [2,3]. It is composed of three ontologies, named Biological Process (BP), Molecular Function (MF) and Cellular Component (CC). Each ontology consists of a set of terms (GO terms) representing different functions, biological processes and cellular components within the cell. GO terms are connected each other to form a hierarchical graph. Terms representing similar functions are close to each other within this graph.

Biological molecules are associated with GO terms that represent their functions, biological roles and localization. This process, usually referred to as annotation process, can be performed under the supervision of an expert or in a fully automated way. Obviously, computationally inferred annotations, commonly known as Electronically Inferred Annotation (IEA), are not as reliable as experimentally determined annotations. For this reason every annotation is labeled with an Evidence Code (EC) that keeps track of the type of process used to produce the annotation itself. Considering the release of annotations of April, 2010, about the 98% of all the annotations is an IEA annotation [4].

The term annotation corpus is commonly used to identify all the annotations involving a set of proteins or genes, usually referring the whole proteomes and genomes (i.e. the annotation corpus of yeast). For lack of space we do not further describe the Gene Ontology. A comprehensive review has been provided by du Plessis et al. [4] and by Guzzi et al. [5].

The availability of well formalized functional data enabled the use of computational methods to analyse genes and proteins from the functional point of view. For example, a set of algorithms, known as functional enrichment algorithms, have been developed to determine the statistical significance of the presence (or the absence) of a GO Term in a set of gene products. A detailed review of these algorithms can be found in [4].

An interesting problem is how to express quantitatively the relationships between GO terms. Several measures, referred to as (term) semantic similarity (SS) measures, has been introduced in the last decade. Given two or more GO terms, they try to quantify the similarity of the functional aspects represented by the terms within the cell. Exploiting annotation

corpora, semantic similarity measures have been further extended to the evaluation of the similarity of genes and proteins on the basis of their annotations.

Many different works have focused on the following tasks: (i) the definition of ad-hoc semantic similarity measures tailored to the characteristics of Gene Ontology; (ii) the definition of measures of comparison of genes and proteins; (iii) the introduction of methodologies for the systematic assessment of semantic similarity measures; (iv) the use of semantic similarity measures in many different contexts and applications. Despite its relevance, the application of semantic similarity for the systematic analysis of protein data is still an open research area. There are, in fact, two main questions that have to be addressed: (i) the systematic assessment of SS with respect to other biological features, i.e. how much an high or a low value of SS is biologically meaningful; (ii) how reliable are the SS themselves, i.e. is there any systematic error or bias in the calculation of SS? Both these problems are relevant for the diffusion of SS measures; while in the first case several approaches have been proposed, confronting SS measures with a pletora of different biological features, only few works dealt with the second problem in a systematic way [5,6,7].

REFERENCES

[1] Mario Cannataro, Pietro Hiram Guzzi, Pierangelo Veltri. Protein-to-protein interactions: Technologies, databases, and algorithms. *ACM Comput. Surv*. 43(1):1, 2010.

[2] Francisco Azuaje, Haiying Wang, and Olivier Bodenreider. Ontology-driven similarity approaches to supporting gene functional assessment. *Proc. of The Eighth Annual Bio-Ontologies Meeting*, pp. 9–10, 2005.

[3] M. A. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, C. Mungall, J. Richter, G. M. Rubin, J. A. Blake, C. Bult, M. Dolan, H. Drabkin, J. T. Eppig, D. P. Hill, L. Ni, M. Ringwald, R. Balakrishnan, J. M. Cherry, K. R. Christie, M. C. Costanzo, S. S. Dwight, S. Engel, D. G. Fisk, J. E. Hirschman, E. L. Hong, R. S. Nash, A. Sethuraman, C. L. Theesfeld, D. Botstein, K. Dolinski, B. Feierbach, T. Berardini, S. Mundodi, S. Y. Rhee, R. Apweiler, D. Barrell, E. Camon, E. Dimmer, V. Lee, R. Chisholm, P. Gaudet, W. Kibbe, R. Kishore, E. M. Schwarz, P. Sternberg, M. Gwinn, L. Hannick, J. Wortman, M. Berriman, V. Wood, P. Tonellato, P. Jaiswal, T. Seigfried, and R. White. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research*, 32 (Database Issue):258–261, 2004.

[4] Louis du Plessis, Nives kunca and Christophe Dessimoz. The what, where, how and why of gene ontologya primer for bioinformaticians. *Briefings in Bioinformatics*, 2011.

[5] Pietro Hiram Guzzi, Marco Mina, Concettina Guerra and Mario Cannataro. Semantic Similarity Measures: Assessment with biological features and Issues. *Briefings In Bioinformatics*, 10.1093/bib/BBR066, 2012.

[6] Da Wei Huang, Brad T. Sherman and Richard A. Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research*, 37(1):1–13, 2009.

[7] Young-Rae Cho, Woochang Hwang, Murali Ramanathan and Aidong Zhang. Semantic integration to identify overlapping functional modules in protein interaction networks. *BMC bioinformatics*, 8:265, 2007.

3 Systems and Prototypes

PathCase-SB: Database-Enabled Tools for Regulatory Metabolic Networks

X. Qi, S.A. Coskun, A. Cakmak, E. Cheng, E. A.Cicek, M. Das, L. Yang, R. Jadeja, S. Syed, R.K. Dash, N. Lai, G. Özsoyoğlu, Z. M. Özsoyoğlu

ABSTRACT

Integration of metabolic pathways resources and metabolic network models, and deploying new tools on the integrated platform is useful for systems biology research on understanding the regulation of metabolic networks. PathCase-SB is such an integrative a web-based application, providing a database-enabled framework and tools towards effective and efficient systems biology model development for mechanistic simulations of biological systems. Current PathCase-SB user interfaces include browser, querying, visualization, provenance, simulation and model composition interfaces. PathCase-SB is built, released, and already being used by researchers across the globe.

INTRODUCTION

PathCase Systems Biology (PathCase-SB) is a web-based application that brings together (a) systems biology data sources (currently BioModels Database [1]), and (b) pathways data sources (currently, KEGG [2]), with the goal of providing additional and new capabilities and tools, made possible due to the integration. The premise is that the integrated use of regulatory metabolic network models and metabolic pathways resources allows for new capabilities and tools to be built that would not be possible otherwise, and thus can help systems biology researchers in understanding the regulation in metabolic networks.

PathCase-SB does not curate models or pathways. It provides (currently, six) user interfaces for users to (i) explore (browse), search and query both models and pathways, (ii) visualize both pathways and modeled networks, and view their mappings in multiple ways, (iii) comparatively simulate, possibly with users' own data, either BioModels Database models or users' own models uploaded to PathCase-SB, (iv) compose a new model from existing SBML models in a semi-automated manner.

In sections below, we summarize the capabilities of PathCase-SB, Version 2, with additional capabilities of model composition added recently. More details about the PathCase-architecture and database design, and PathCase Interfaces are available elsewhere [3, 4].

METHODS AND IMPLEMENTATION

PathCase-SB Database

The database (a Microsoft SQLServer database) is designed to contain data from different systems biology and biochemical network databases in separate tables for common entities of different (systems biology and/or biochemical network) data sources, e.g., currently, BioModels Database (and, in the future, CellML [5]) and KEGG (and, in the future, Reactome [6]) data sources. As an example, we maintain a species table occurring in BioModels Database models and molecular_entities table for KEGG molecular entities. Such an approach allows us to separate data from different data sources cleanly, and, to add new data sources seamlessly, without dealing with data cleansing, data integration, and data curation problems.

Strict data-separation-per-data-source approach of PathCase-SB also requires an additional "mapping" effort involving all pairs of data sources on their corresponding entities. For BioModels Database and KEGG, this involves three mappings: <species, molecular entities>, <reactions, process-entities>, <models, pathways>. PathCase-SB database also keeps separate tables for different systems biology data sources, e.g., currently, distinct sets of tables are maintained for BioModels Database and CellML (not yet open to public) data sources. Please see [3] for more details.

Currently, PathCase-SB database contains four classes of tables capturing the following information: (a) biochemical reaction network-related tables, (b) systems biology data source-related tables (e.g., quantitative kinetic information, dynamic behavior, involved species, reactions, etc.), (c) tables mapping data from different data sources, (d) tables annotating systems biology data with other ontologies or taxonomies, e.g., Gene Ontology [7].

PathCase-SB Browser Interface

Browser interface provides a variety of browsing-based mechanisms for users to access the PathCase-SB database, starting from a basic overview that lists the entities in the database to hierarchically drilled-down levels that include, among others, reaction, species, and compartments. The PathCase-SB browser interface presents a unique multi-faceted view of the database, which allows users to access and organize the biochemical information with distinct focus points. As an example, researchers have the option to browse models by their corresponding pathways, studied organisms, or relevant GO terms (e.g., for an enrichment pre-study). Each browser item is linked to an information-rich "details page" that organizes (i) lists of participants and their roles in each model entry, and the kinetic models of the corresponding biochemical reactions and their parameters, (ii) gateways to interactive graphical tools and interfaces (e.g., simulation and visualization engines), (iii) data provenance information for source tracking, and (iv) entry points to related parameterized queries for a customized and focused study of the underlying data. Fig.1, part A illustrates the browser expansions as a result of clicking (i) to "BioModels Database models by name" (leftmost pane), and, then (ii) to model *Akman2008_Circadian_Clock_Model2* (next leftmost pane), which results in the final expansion.

The PathCase-SB Browser interface provides to users

- An embedded in-place keyword search facility with paged result listings,
- Relationships between BioModels Database models and ontologies (e.g., Gene Ontology, EC (Enzyme Commission) number taxonomy [8], and System Biology Ontology [9]),
- Biological compartment-based relationships between different models. The idea is to allow

modelers to see the listings of models that capture biological networks within the same compartment (e.g., say, liver cytosol) to help them with larger-model composition and model merge operations.

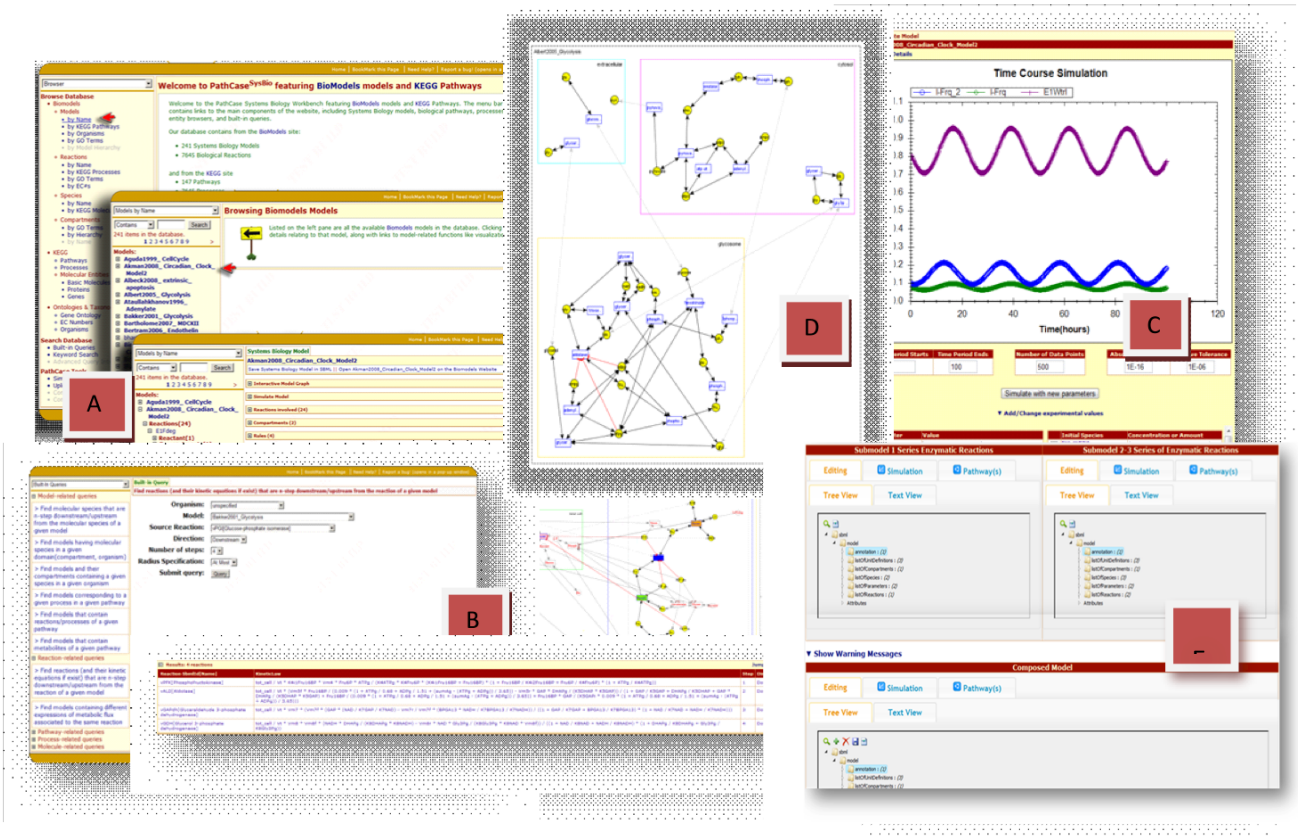

Figure 1. Sample views of the PathCase-SB System. Parts A, B, C, D, and E represent Browser Interface, Built-In Queries Interface, Simulation Interface, Visualization

PathCase-SB Querying Interface

PathCase-SB is designed for users to pose built-in (i.e., predefined) queries involving models and other database objects. Built-in queries can be characterized as a small set of popular queries that are provided through very simple user interfaces. Note that some of the built-in queries are pathways-only queries from PathCase[10], and, provided only as a convenience.

PathCase-SB built-in queries are grouped into

- *Model queries*. E.g., "Find models and their compartments containing a given species in a given organism".
- *Pathway queries*. E.g., "Visualize a set of pathways".

Reaction queries. E.g., "Find reactions (and their kinetic equations if exist) that are n-step downstream/upstream from the reaction of a given model", which is illustrated in Fig. 1, part B: (i) query specification shown in the pane to the left of "B", (ii) visualized query result shown in the pane to the right of "B", and (iii) textual query result shown in the pane below "B".

- *Species/molecule queries*. E.g., "Find species that are n-step downstream/upstream from the species of a given model").

In the spirit of the open communication framework, all built-in queries are made available through

web services, so that third-party applications can directly send their query execution requests to the web services, execute the requested PathCase-SB built-in query, and receive the execution results as an SBML (for a model-based output) or as a BioPax (for a metabolic network graph output) document. Such an approach promotes open data exchange, and is beneficial to other tool builders. This is also consistent with the currently available web services for built-in queries of the existing PathCase system.

PathCase-SB Simulation Interface

This interface allows users to either directly use the SBML files of models in the PathCase-SB database or to upload their own SBML files of models (the *iModel Tool*), and simulate them. In addition, users can see simulation results side by side in a comparative manner (the *SimCom Tool*). For the simulation, a high performance cellular network simulation service, namely, RoadRunner [11, 12] is used through the REST web services provided by Systems Biology Workbench application[13] programming interface (API). A third party library called ZedGraph (available at http://zedgraph.org) is employed to render the simulation results as a graph.

Fig. 1, Part C illustrates the simulation of the model *Akman2008_Circadian_Clock_Model2* (also shown in Fig. 1, Part A). Note that the simulation interface lets users to modify: a) parameter values; b) initial conditions for time, concentrations or amounts of species and boundary species; c)number of data points to plot; d) absolute and relative tolerance. Also, it allows to select metabolic fluxes to plot, , add experimental values, and visualize results of new simulation within the same time-course simulation graph.

Within the simulation interface, input for experimental results is manually editable on the field specified for users. Users can directly use comma or space delimited input from commonly used environments such as MATLAB or Excel. Users can also access model details (e.g., model version) from the link above the simulation graph.

In general, there is more than one mathematical model for the same pathway. Therefore side-by-side comparisons of model simulations for the same pathway can allow researchers to observe similarities and differences between models. The *SimCom tool* provides the functionality to simulate up to four different models in the same pathway side by side (in new pop-up windows) from PathCase Systems Biology web site.

The *iModel tool* allows users to upload their own SBML models onto the PathCase Systems Biology web site to simulate. Uploaded models are parsed by the *PathCase SBML Parser* [14] which uses libSBML[15] library. After parsing, the model is stored in a separate temporary database (which is emptied regularly for privacy and copyright protection purposes; therefore the uploaded models are not kept in our real database), and input to the *iModel tool*.

In terms of import/export capabilities, PathCase-SB does not provide export capabilities for models and pathways since all of its models and pathways are available in the original data sources (e.g., BioModels Database, KEGG). Import capabilities for pathways is provided by PathCase[10], and import capabilities for models is provided by the *iModel tool*.

Finally, simulation interface is fully-integrated with model detail pages to simulate a particular model in-place, while allowing browsing model details.

PathCase-SB Visualization Interface

A client side java applet, called PathCase-SB Graph Viewer, produces interactive pathway graphs, model graphs, or both with various mappings from one to the other. The visualized model/pathway

can be manually or automatically rearranged, zoomed in/out, panned, expanded/collapsed, queried from, saved locally as jpeg file, and studied in detail. Fig. 1, part D, illustrates the visualization of the BioModels Database systems biology model *Albert2005_Glycolysis* (reached by clicking to the "Interactive Model Graph" link of the model). Note that the modelers of this model have specified three compartments, namely, "extracellular", "cytosol", and "Glysome". Also note that Pathcase-SB system does not curate data, and simply displays the model/pathway as curated by the original data source (i.e., in this model's case, the BioModels Database).

All visualization features are provided on client-side, and without server-side communication, allowing for high scalability. Visualization is embedded into the corresponding model/pathway pages, and (i) does not require any separate installation effort as a manageability convenience for users, and (ii) provides a platform-independent access regardless of the client operating system or browser differences (presently optimized for IE, Firefox, Safari, and Chrome browsers).

PathCase-SB *Graph Viewer*, when accessed from different places within PathCase-SB, has many different legends, basic controls, and toolbar capabilities. The *Graph Viewer* is employed by

- *Browser Interface* (appears as a menu item at many places with name "Interactive Model/Pathway Graph") to visualize the full PathCase-SB metabolic network (in multiple condensed/expanded forms), individual pathways, metabolic subnetworks, and networks of systems biology models,
- *Built-In Queries*: For each query that produces a metabolic subnetwork, the results are visualized by the *Graph Viewer*,
- *The iModel Tool* (e.g., "Upload your own model"): Biochemical networks of user models are visualized by the *Graph Viewer*.

The functionality of the *Graph Viewer* will be expanded in the future, to be used in future tools "Compose-Models" and "Compare-Models".

PathCase-SB Provenance Tool

Provenance, also called lineage or pedigree, is defined as metadata that tracks the steps of data derivation, which adds value to the data itself [16]. With our use of other web-based data sources, providing provenance of data in PathCase-SB is a necessity. We provide three levels of provenance data:

- *Model related information:* Creation date, Modification date, Notes of author and publication id. This data is stored in our servers.
- *Authors and publications:* Author-related information is stored in our servers, whereas publication data is referred to the outer source (BioModels Database CiteXplore),
- *Papers* that are being cited by the publication of the model. This level is again hosted by BioModels Database CiteXplore.

All three layers of provenance are specified via a stand-alone panel: when a model in a panel is opened, provenance data about the selected model is displayed.

Finally, as a short comparison, we note that there are at least three web-based applications with model simulation and visualization features (http://cytosolve.mit.edu, http://jjj.biochem.sun.ac.za, and http://www.itb.cnr.it/cellcycle). However, PathCase-SB is a comprehensive web-based application that not only includes simulation and visualization (of model networks and pathways), but also has database, browser and querying interfaces, as well as a provenance tool.

Model Composition Interface

This interface is an AJAX[17] enabled advanced web-based biological model composition interface for SBML formatted mathematical models. The interface allows user to choose two standard SBML models, either from the user's computer or from PathCase-SB database, and merge them into a new model. Part of the merge process is completed automatically and then the modeler can manually alter combined models to finalize the merge results. Fig. 1, Part E illustrates an overview of this interface.

Within the model composition interface, four components, namely, Tree View, Text View, Simulation and the Pathway(s) visualization tabs are provided. The tree and text views enable the user to edit the SBML model. The Tree View contains an easy-to-use hierarchical XML representation of the SBML model. The Text View gives advanced users the ability to go over the original SBML file. The Simulation and Pathway(s) visualization tabs have the same capabilities as discussed before. Both source modes and new merged models can be simulated and visualized in the integrated model composition interface.

Currently, SBML versions 1 and 2 of Level 1, Version 1, 2, 3 and 4 of Level 2, and Version 1 of Level 3 are all supported. For more details about interfaces and/or architectural details about PathCase-SB model composition interfaces, please see [18].

Future Extensions

In the future, we are planning to

- add CellML models and Reactome pathways into PathCase-SB, and
- add to PathCase-SB visualization interface special features that allow users to provide "omics" related data analysis.

ACKNOWLEDGEMENT

We gratefully acknowledge the ideas and contributions of our deceased co-researcher Dr. Marco Cabrera.

REFERENCES

[1] N. Le Novère, et al, "BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems." *Nucleic Acids Res*. 34 (Database-Issue), 2006: 689-691.

[2] M. Kanehisa, et al, "From genomics to chemical genomics: new developments in KEGG." *Nucleic Acids Res*. 34, 2006, D354-357.

[3] A. Cakmak, et al, "PathCase-SB architecture and database design." *BMC Syst Biol*. Nov 2011 9;5:188.

[4] S. Coskun, et al , "PathCase-SB: Integrating Data Sources and Providing Tools for Systems Biology Research." *BMC Syst Biol*. , submitted for publication.

[5] C.M. Lloyd, et al. "CELLML: Its future, present, and past." *Progress in Biophysics and Molecular biology*. Vol. 85, Issues 2-3, 2004, pp. 433-450.

[6] L. Matthews, et al, "Reactome knowledgebase of human biological pathways and processes."

Nucleic Acids Res. Jan 2009;37(Database issue):D619-22.

[7] M. Ashburner, et al, "Gene Ontology: tool for the unification of biology." Nat. Gen. May 2000, 25(1):25-9.

[8] E. C. Webb, *Enzyme Nomenclature 1992*. Int. Union of Biochemistry and Molecular Biology, Academic Press.

[9] N. Le Novere, et al, "Adding semantics in kinetic models of biochemical pathways." *ESCEC Conf*., 2006.

[10] B. Elliott, et al, "PathCase: Pathways database System. Metabolic Pathways System." *Bioinformatics*. Vol. 24, 2008, No. 21, pp.2526-2533.

[11] F.T. Bergmann, and H.M. Sauro, " SBW – a modular framework for systems biology." in *Proc., 38th Conf. on Winter Simulation*, 2006, pp. 1637–1645.

[12] F.T. Bergmann, and H.M. Sauro, "Comparing simulation results of SBML capable simulators." *Bioinformatics*, Vol. 24, 2008 Pp. 1963 – 1965.

[13] H. M. Sauro, et al. "Next generation simulation tools: the Systems Biology Workbench and BioSPICE integration." *OMICS*. Winter, 2003; 7(4):355-72.

[14] M. Das, "PathCase: SBML Parser and CellML Parser." MS project. CWRU 2010.

[15] J. B. Benjamin, et al, "LibSBML: an API Library for SBML." *Bioinformatics* 24(6), 2008: 880-881.

[16] R. Bose, and J. Frew, "Lineage retrieval for Scientific Data processing: a survey." *ACM Comp*. 2005, 37(1).

[17] http://en.wikipedia.org/wiki/AJAX

[18] A. Coskun, "PathCase-SB Model Composition Interface: Biological Model Composition Tool for SBML Models", manuscript under preparation, 2012.

3. Conference Announcements

ACM Conference on Bioinformatics, Computational Biology Biomedicine (ACM-BCB 2012): Call for Participation

Conference Dates: October 7-10 2012
Conference Venue: Embassy Suites, Downtown Orlando, FL
Conference Website:http://www.cse.buffalo.edu/ACM-BCB2012/

You are invited to attend ACM Conference on Bioinformatics, Computational Biology and Biomedicine (ACM BCB). ACM BCB is the main flagship conference of the ACM SIG Bioinformatics. ACM BCB 2012 is in its third year, building upon the success of ACM BCB 2010 in Niagara Falls and ACM BCB 2011 in Chicago. Each of the conference had about 200 attendees.

PROGRAM: The ACM BCB 2012 technical program, including a list of accepted papers, is available athttp://www.cse.buffalo.edu/ACMBCB2012/acceptedPapers.html

The program includes invited talks by Martha L. Bulyk (Harvard), Ying Xu (Georgia) and Pierre Baldi, (UC Irvine). There are three tutorial and four workshops (http://www.cse.buffalo.edu/ACM-BCB2012/workshops.html).

WORKSHOPS: Four workshops are planned to be held in conjunction with ACM-BCB 2012:

1. Immunoinformatics and Computational Immunology Workshop (ICIW 2012)

2. Biological Network Analysis and Applications in Translational and Personalized Medicine (BNA-M 2012)

3. 1st International Workshop on Parallel and Cloud-based Bioinformatics and Biomedicine (ParBio 2012)

4. 1st International Workshop on Robustness and Stability of Biological Systems and Computational Solutions

Call for Contribution

Dear SIGBioinformatics member,

The aim of SIGBioinformatics Record is to report on the activities conducted by the community, and to publish early notes on current progress in bioinformatics and biomedical informatics areas.

In addition, special articles will be published highlighting important developments in the relevant fields.

An important place in our newsletter will be devoted to collect description of different research domains of the members.

As Bioinformatics and Computational Biology is now a broad research area, SIGBioinformatics Record could be a mechanism to publicize problem domains.

Nevertheless SIGBioinformatics Record could stimulate discussions and bridge different areas.

Thus we kindly invite you to outline your research area (description should not exceed two pages, following the attached template – latex template could be found on the Guidelines on the ACM Website).

Description will be published on the newsletter (see the ACM Digital Library for the latest issue),

Moreover, you can send us a brief description of "student-activities", i.e. ongoing student projects and educational activities carried out.

Important Dates

- Publication dates: 30th January, 30th May, 30th September
- Due dates: 1st January, 1st May, 1st September.