BioStar models of clinical and genomic data for biomedical data warehouse design

Liangjiang Wang and Aidong Zhang*

Department of Computer Science and Engineering,
State University of New York at Buffalo,
201 Bell Hall, Buffalo, NY 14260, USA
Fax: 716 645 3464 E-mail: ljwang2@cse.buffalo.edu
E-mail: azhang@cse.buffalo.edu
*Corresponding author

Murali Ramanathan

Department of Pharmaceutical Sciences,
State University of New York at Buffalo,
Buffalo, NY 14260, USA
Fax: 716 645 3693 E-mail: murali@acsu.buffalo.edu

Abstract: Biomedical research is now generating large amounts of data, ranging from clinical test results to microarray gene expression profiles. The scale and complexity of these datasets give rise to substantial challenges in data management and analysis. It is highly desirable that data warehousing and online analytical processing technologies can be applied to biomedical data integration and mining. The major difficulty probably lies in the task of capturing and modelling diverse biological objects and their complex relationships. This paper describes multidimensional data modelling for biomedical data warehouse design. Since the conventional models such as star schema appear to be insufficient for modelling clinical and genomic data, we develop a new model called *BioStar* schema. The new model can capture the rich semantics of biomedical data and provide greater extensibility for the fast evolution of biological research methodologies.

Keywords: clinical and genomic data integration; multidimensional modelling; data warehouse design.

Reference to this paper should be made as follows: Wang, L., Zhang, A. and Ramanathan, M. (xxxx) '*BioStar* models of clinical and genomic data for biomedical data warehouse design', *Int. J. Bioinformatics Research and Applications*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Liangjiang Wang is a Postdoctoral Fellow in the Department of Computer Science and Engineering, State University of New York, at Buffalo. He received his PhD from the University of Georgia, majoring in molecular biology. He also received an MS degree in computer science from the Mississippi State University. His research interests include biological database design, data analysis, and computational biology. His work has been published in journals including *Bioinformatics, Genetics, Plant Cell, Plant Physiology, PNAS* and *Theoretical & Applied Genetics*.

Aidong Zhang is a Professor in the Department of Computer Science and Engineering at State University of New York at Buffalo. Her research interests include multimedia, bioinformatics, and data mining. She is an author of over 150 research publications in these areas. Zhang serves on the editorial boards of *International Journal of Bioinformatics Research and Applications* (IJBRA), ACM Multimedia Systems, the *International Journal of Multimedia Tools and Applications*, and *International Journal of Distributed and Parallel Databases*. She has also chaired or served on various conference program committees. Zhang is a recipient of the National Science Foundation CAREER award and SUNY Chancellor's Research Recognition award.

Murali Ramanathan is Associate Professor in the Departments of Pharmaceutical Sciences and Neurology at the State University of New York at Buffalo. He received his BTech (Hons) in chemical engineering from the Indian Institute of Technology, Kharagpur, India, an MS in chemical engineering from the Iowa State University, Ames, IA, and a PhD in bioengineering from the University of California, San Francisco, CA. His research focuses on the clinical pharmacogenomics of multiple sclerosis and on the application of bioinformatics and mathematical modelling techniques to clinical and pharmaceutical problems.

1 Introduction

Recent developments of high throughput technologies result in large datasets of genomic sequences and gene functional profiles. Analysis of these datasets may lead to greater understanding of the biological mechanisms behind diseases such as multiple sclerosis. For example, microarray data contain valuable information for discovery of disease-associated gene expression patterns and classification of patients (Golub et al., 1999; Ramaswamy et al., 2001). The scale and complexity of genomic datasets give rise to substantial challenges in data management and mining. It is also clear that full benefit of functional genomics may only be obtained through seamless integration with clinical data and biological background knowledge. However, the diverse resources of clinical and genomic information are typically distributed at a range of sites. These information resources often allow the data to be browsed or downloaded as flat files, but do not support efficient genome-wide data analysis and integration with other data sources of interest.

Data warehousing technology, which was originally developed in a business context, is beginning to be used in the fields of biology and medical sciences to meet the requirement of data mining for clean and consistent data. A data warehouse is defined as "a subject-oriented, integrated, non-volatile and time-variant collection of data in support of management's decisions" (Inmon, 1996). Specifically, business data are extracted from several operational databases, transformed, cleansed and loaded into a multidimensional database. The data in the warehouse may be further filtered, aggregated and stored in smaller data stores, usually called data marts, for specialised purposes. Thus, data warehouses are viewed as consolidated repositories of historical data and their major role is to facilitate business decision making.

Online analytical processing (OLAP) applications are built to provide users a multidimensional view of the data in the warehouse and to allow data analysis through ad hoc queries of the data (Cabibbo and Torlone, 1998). An important feature of OLAP is the presentation of information at different levels of detail through aggregating and disaggregating data over one or more dimensions (Marcel, 1999). This feature is realised by two OLAP operations, called roll-up and drill-down. Roll-up corresponds to summarisation of data for the next higher level of a concept hierarchy associated with a dimension. Drill-down, which is the reverse of roll-up, provides navigation from a higher-level summary to the lower-level detailed data. Other OLAP operations include pivoting, slicing and dicing (Vassiliadis, 1998). Since measurable business facts or the so-called measures are mostly numeric values (e.g., the dollar amount of a sale in a retail business), roll-up usually uses simple aggregate functions such as sum and average, though complex algebraic or statistical operators may also be defined for OLAP operations (Datta and Thomas, 1999).

While data warehousing and OLAP have been successfully applied to the business domain, it is clear that direct transfer of these technologies to biology is fraught with difficulties (Dubitzky et al., 2001). The main reason is that the information need of biological research is fundamentally different from that of customer-centred business. While business data analysis such as market-driven trend analysis is to support management's decision, the main goal of biological data warehousing is probably to provide a global and integrated view of living systems. Another major difficulty is due to the great complexity of biology. Unlike business processes that are logically simple and temporally stable, biology has very complex research methodologies and a huge fast-growing body of background knowledge. The task of capturing, modelling and encoding some of the biological knowledge for a data warehouse appears to be a great challenge.

Although there is a pressing need for robust multidimensional models of biological data, only a few papers have been published in this area. Markowitz and Topaloglou (2001) developed conceptual models for multidimensional analysis of microarray gene expression data. The modelling data spaces included sample, gene annotation and gene expression. The authors proposed to use star or snowflake schemas for logical design, though the logical data models were not specified in detail. In addition, various clinical and genomic data were not modelled. Pedersen et al. (2001) investigated the structure of some clinical data and developed a multidimensional data model that extended star schema. Although some of the challenging problems of clinical data modelling were taken into consideration, the extended data model was not comprehensive and did not cover microarray gene expression and other genomic data.

Biological data are also modelled using object-oriented and traditional entity-relationship (ER) approaches. Paton et al. (2000) described a collection of object-oriented conceptual models for various yeast data, including genetic, genomic sequence, gene expression and protein-protein interaction data. The models were later implemented in the object database called GIMS (Cornell et al., 2001). Many others used traditional ER models for microarray gene expression and/or other genomic data (Gollub et al., 2003; Chen et al., 2004). These databases were aimed to provide management and integration of genomic data, and thus support for efficient multidimensional data analysis was probably not considered in the database design.

We present in this paper a new model for clinical and genomic data in the warehouse design. In our case, the data warehouse is designed to integrate various biomedical datasets for studies of human diseases (e.g., multiple sclerosis and cancers). Clinical data, including clinical test results, MRI images and drug responses of patients are provided by our collaborators. A relatively large collection of microarray gene expression data from multiple sclerosis patients is available for this research. Other genomic datasets, including sequences and annotations, microarray gene expression data, protein-protein interaction data and protein domain information, are obtained from public databases. Thus, our data modelling scope is quite comprehensive and the work may provide a general framework for biomedical data warehousing and mining.

The rest of this paper is organised as follows. Section 2 provides several examples of modelling clinical and genomic data at the conceptual level. We discuss new challenges for the multidimensional modelling of biomedical data in Section 3. In Section 4, a new model called *BioStar* is described. We then use *BioStar* to model complex biomedical data spaces in Section 5.

2 Motivating examples

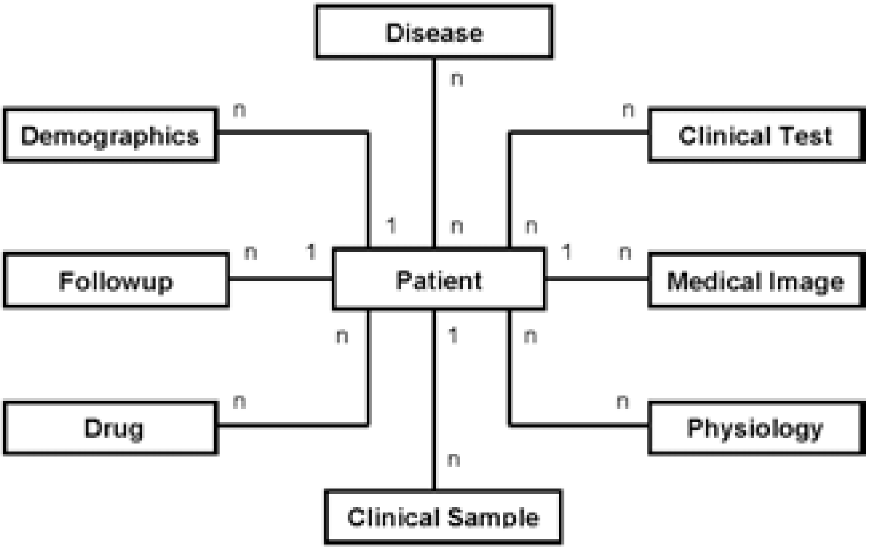
The data warehousing and mining lifecycle includes data modelling, data warehouse construction, and development of visualisation and mining tools. The data models should provide a multidimensional view of data and serve as a foundation on which a data warehouse can be built. The data modelling process may be divided into three different design phases: conceptual data modelling deals with high-level representation of the data space; logical data modelling relates high-level concepts to a certain kind of database management system (DBMS); and physical design specifies how data are actually stored using a specific DBMS.

In this section, we describe three typical examples of modelling clinical and genomic data at the conceptual level. Due to the diversity and complexity of biomedical data, the warehouse design may include several modelling data spaces. In our work, we used the following six data spaces: clinical data space, sample data space, microarray data space, proteomic data space, experiment data space, and gene data space.

Example 1 (clinical data space): An entity-relationship (ER) diagram for the clinical data space is shown in Figure 1. An entity is represented as a rectangle with the entity name and a relationship between two entities is drawn using a line with the multiplicity label, which indicates the number of objects that may participate in the relationship. For example, one Patient can have many (denoted by ‘*n*’) Clinical Samples taken for laboratory assays (Figure 1). The attributes of the entities are not shown in the conceptual models.

The clinical data space has a rich variety of entities, among which Patient is obviously the most important entity and thus the fact entity in the multidimensional data model. All the other entities can be viewed as dimensions to characterise patients. Disease and Drug represent two important dimensions of patient data. Both dimensions have a many-to-many relationship with Patient and their associated measures, namely disease diagnosis and drug use, which need bi-temporal support to specify their valid time intervals. In addition, the relationship between Disease and Patient can be uncertain in some cases.

Figure 1 A conceptual model of the clinical data space

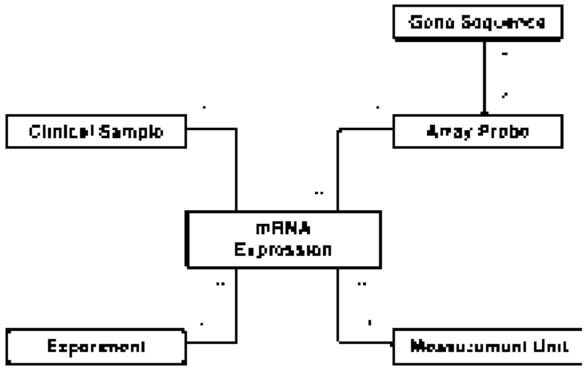


Clinical Sample is another important entity, which has a many-to-one relationship with Patient. Clinical samples such as blood samples are taken from patients and used for various laboratory assays. The Clinical Test entity captures information about simple clinical tests applied directly to patients through physical examination or by asking patients to perform some simple routines. Clinical Test has a many-to-many relationship with the Patient entity. Patient data from sophisticated clinical studies are captured by the Medical Image and Physiology entities. These studies use advanced medical equipment and result in complex data such as MRI images and cardiograms. The Medical Image entity has a many-to-one relationship with Patient, whereas the Physiology entity has a many-to-many relationship with Patient. Other dimension entities in our design include Demographics, which characterises patients based on demographic information and Followup, which captures patient status information in followup reports. Both entities have a many-to-one relationship with Patient.

The clinical data space has a very complex structure and may include more entities such as patient family history record. These dimensions characterise patients with different fact measures. However, for a particular patient, only a few measures are usually available.

Example 2 (microarray data space): The DNA microarray technologies allow genome-wide analysis of gene expression at the mRNA level. In Figure 2, a conceptual model for the microarray data space is shown. The fact entity in this data space is mRNA Expression, which has four dimension entities. The Clinical Sample entity provides sample information for gene expression measurements and it links the microarray data space with the clinical data space described above.

Figure 2 A conceptual model for the microarray data space



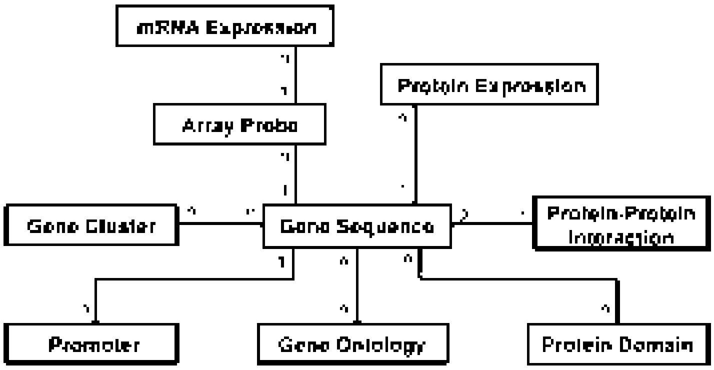
Array Probe captures the information about sequence or oligonucleotide probes that are placed on the microarray. These probes are derived from gene sequences. Since multiple probes may be used for a single gene, it is often necessary to summarise gene expression to the higher level of non-redundant gene sequences.

The Measurement Unit entity can be used to keep information about what is measured for gene expression. For example, the Affymetrix GeneChip platform provides two kinds of gene expression measurements for each probe set, a presence/absence (PA) call and a numeric value. In contrast, cDNA microarrays give the relative ratios of gene expression in two samples.

The Experiment entity captures metadata of each experiment. Microarray data are obtained through complex experimental processes. The procedures and platforms used for such experiments can affect the magnitude and quality of gene expression measurements. Since datasets generated in different research laboratories and possibly using different platforms will be combined in the data warehouse, it is very important to capture the experimental metadata for both data quality control and future reference.

Example 3 (gene data space): The gene data space contains gene function information integrated from a variety of public domain data sources. Since NCBI's non-redundant RefSeq or UniGene sequence dataset is normally used as the reference set for functional annotations, we view Gene Sequence as the fact entity of the gene data space (Figure 3).

Figure 3 A conceptual model for the gene data space



There are many possible dimensions to characterise the gene functions and seven of these entities are shown in Figure 3. The entity important for microarray data analysis is Array Probe, which has a many-to-one relationship with Gene Sequence. The Array Probe entity links the gene data space to microarray data space. Protein Expression represents another level of gene expression measurement using proteomic approaches and has a many-to-one relationship with Gene Sequence. In the proteomic data space, Protein Expression is the fact entity (similar to mRNA expression in the microarray data space).

The other gene data dimensions include Gene Cluster, Promoter, Protein Domain and Protein-Protein Interaction. Co-regulated gene clusters are obtained by clustering analysis of gene expression data in the warehouse and can be used to analyse gene regulatory networks, together with information about promoters and their composition of sequence motifs. Protein interaction and domain information are two important dimensions for annotating gene functions.

3 Problem description

One of the major requirements in operational database design is to avoid data redundancy in database relations through normalisation. Normalised schemas are important for supporting efficient online transaction processing (OLTP), which include pre-defined statements of queries and updates. However, the data access characteristics of a data warehouse are quite different from those of operational databases. In a data warehouse environment, users do not initiate update transactions. On the other hand, a data warehouse needs to support complex ad hoc queries that compute aggregate values over a huge amount of data for the purpose of data analysis such as OLAP. If a data warehouse uses a standard Entity-Relationship schema with many tables, a user may not be able to fully understand and utilise the integrated information in the warehouse (Kimball, 1996). Thus, data warehouse design often adopts multidimensional data modelling, which organises database entities into facts and dimensions.

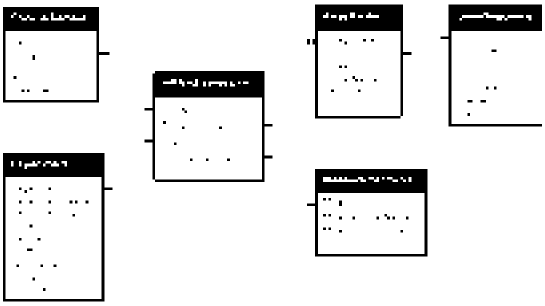
3.1 Application of existing multidimensional models

The application of multidimensional models to biomedical data warehousing is a recent effort. Most of the multidimensional models documented in the literature are based on business data (Abello et al., 2001) and thus may not meet the requirements of biomedical data warehousing. Pedersen et al. (2001) evaluated 14 existing multidimensional models against the requirements of clinical data warehousing and found that none of these models support all the requirements.

In a relational online analytical processing (ROLAP) architecture, a star or snowflake schema is commonly used for the data warehouse design (Vassiliadis and Sellis, 1999). A star schema consists of one central fact table, which stores measures for OLAP aggregation and several denormalised dimension tables. The major advantages of star schemas are their support for efficient OLAP operations and high understandability to warehouse users. The normalised version of a star schema is called a snowflake schema, in which dimension hierarchies can be explicitly defined using separate tables. If there are multiple fact tables to share dimension tables, such a collection of star schemas is called a fact constellation.

We are building our biomedical data warehouse using a relational database management system (RDBMS). Our data modelling work suggest that star or snowflake schemas may be used to model semantically simple data spaces including the microarray and proteomic data spaces. These data spaces represent well-defined experimental processes and contain many-to-one relationships between facts and dimensions. For example, Figure 4 shows the snowflake representation of the microarray data space. The schema has mRNAExpression as the central fact table, which has many-to-one relationships with its four dimension tables. The mRNAExpression fact table stores the measure of gene expression using the field 'Expression'. The measure can be absolute gene expression values or PA calls from the Affymetrix GeneChip platform. The type of the measure is specified using MeasureUnitID, which is the primary key of the MeasurementUnit dimension table and a foreign key in the fact table (Figure 4). The other three tables (ClinicalSample, ArrayProbe and Experiment) store information about the respective dimensions as described in the previous section. Note that Experiment is also the fact table of the experiment data space and ArrayProbe relates mRNAExpression with GeneSequence, the fact table of the gene data space (see below).

Figure 4 A snowflake schema for the microarray data space



However, star schemas do not appear to be sufficient for modelling the semantics of complex data spaces such as the clinical data space (Figure 1) and gene data space (Figure 3). These data spaces have several features that introduce new challenges to multidimensional data modelling.

3.2 New challenges

We now discuss the major characteristics of clinical and genomic data and compare them with business data to show the different requirements between biomedical and business data warehousing (Table 1). First, the structure of clinical and genomic data is very complex and fast evolving, which reflects the great complexity of biological research and constant advances of experimental approaches. Specifically, many current entity types can be defined as dimensions in both clinical and gene data spaces and even more dimensions may be added over time. Furthermore, each dimension often has its own fact measures, which are coupled loosely with and can change independent of other dimensions' measures. For example, a patient may have clinical tests with associated test results (clinical test measures) and may also be given one or more drugs, giving rise to the drug usage measures. However, the clinical tests may or may not be directly related to the drug usage.

Table 1 Characteristics of clinical and genomic data when compared to business data

Clinical and genomic data	*Business data*
Complex data structure with many potential dimensions	Easy-to-understand data structure with few dimensions
Often many-to-many relationships between facts and dimensions	Many-to-one relationships between facts and dimensions
Uncertain relationships between fact and dimension objects	Certain relationships between fact and dimension objects
Some clinical measures require advanced temporal support for time validity	Historical data, no advanced temporal support needed
Incomplete and/or imprecise data very common	Few incomplete and/or imprecise data

Second, many-to-many relationships between fact and dimension objects are common in the clinical and gene data spaces. For example, a patient can be treated with one or more drugs and a drug can be used by many patients. A gene can have several protein domains, and a domain can be present in many genes. These natural many-to-many relationships are not easily modelled using star schemas, which are originally designed to handle the many-to-one relationships between a business fact and a dimension.

Third, the relationships between fact and dimension objects may be uncertain in some cases. For example, a gene can be annotated to have some functions using gene ontology terms based on currently available evidence. However, the functional annotation may be completely changed when new evidence becomes available. Similarly, the diagnosis of a patient, i.e., the relationship between the patient and diseases, may be uncertain at a given time point, depending on the available clinical test results.

Fourth, one important property of clinical data is that some measures are only valid in specific time intervals. For example, a patient may have a disease at a specific time interval and the effect of a drug on patients often lasts a certain time period. Thus, some clinical data need bi-temporal support (starting and ending time points) to specify their valid time intervals. This advanced temporal concept, although very important for multidimensional analysis of clinical data, is not supported by conventional models.

Finally, clinical and genomic datasets are often incomplete and the values can be imprecise. For example, the data of a particular patient can be a few clinical test results, with no other measures available. The imprecision of data often results from imperfect understanding of biological objects. For example, a gene may only be assigned to a high-level functional category, but its exact function is still unknown.

The above characteristics of biomedical data should be considered as the critical requirements of the multidimensional data model for the biomedical data warehouse. The existing multidimensional models do not fully support these requirements. For example, the clinical data space features complex relationships between the fact entity (Patient) and dimension entities and need bi-temporal support for some clinical measures. A conventional star or snowflake schema has problems to model the clinical data space. If a single fact table is used to store all the different clinical measures, most entries (including foreign keys of the fact table) would contain null values due to the incompleteness of data.

For the gene data space (Figure 3), extensibility of the model is one of the major requirements. Due to the fast evolution of genomics, new dimensions may need to be added to the model. The conventional models do not support this requirement. For example, introduction of a new dimension into a star schema would require re-computing all the data entries in the fact table.

4 *BioStar*: a new model for biomedical data

In Section 4, we describe a new multidimensional model, called *BioStar*, which supports all the requirements of our biomedical data warehousing. We first describe the basic elements of the model and then discuss its properties important for modelling complex biomedical data spaces.

4.1 The basic model

A *BioStar* fact schema is a quadruple $F = (C, D, M, S)$, where C is the central entity schema, D is a set of dimension schemas, M is a set of measure schemas and S is a set of summarisability constraints. The central entity is viewed as a special dimension, which is associated with every fact measure. For example, in the clinical data space, Patient is the central entity; and in the gene data space, Gene Sequence is the central entity.

C or $D_i \in D$ is a pair $(L, \prec)$, where L is a set of dimension levels and $\prec$ is a partial order of the elements in L. A level $l \in L$ is a dimension attribute, which is associated with a domain of values, dom(l). For example, the Demographics dimension of Patient (Figure 1) may have the following levels: Street Address, City, County, State and Country. The dimension levels specify the granularities that can be used to represent the fact objects. For example, patients may be classified by their living places in an increasing order of granularities: Street Address $\prec$ {City; County} $\prec$ State $\prec$ Country. Note that it is a partial order since there are two possibilities of ordering City and County (Street Address $\prec$ City $\prec$ County; or Street Address $\prec$ County). Therefore, $(L, \prec)$ is a lattice specifying the classification hierarchy of C or $D_i \in D$. The classification hierarchy determines how fact objects may be summarised over the dimension. In the biomedical data warehouse, however, some dimension hierarchies are still undefined or very complex. Definition of standard hierarchies for these dimensions requires significant background knowledge of biology (e.g., gene ontology and disease ontology). In this study, we are focused on modelling the structures and semantics of biomedical data and assume that some kind of classification hierarchy exists for these dimensions.

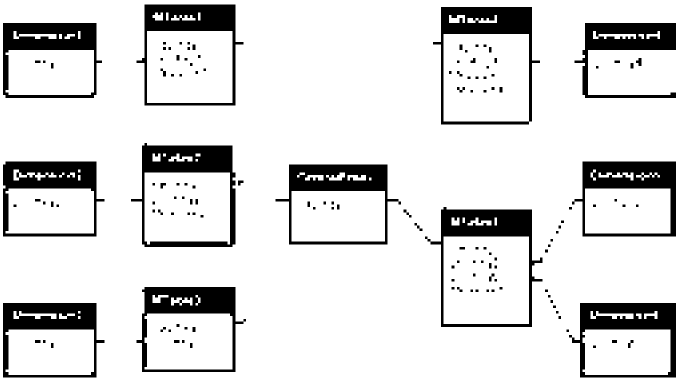
A measure schema, $M_j \in M$, is a triple (A_m, A_s, D_m), where A_m is a set of attributes called measures, A_s is a set of supporting attributes for the measures (e.g., single- or bi-temporal support) and D_m is a set of dimensions that are associated with the measures. In a relational database, a measure schema may be implemented as a separate table, which we call it an *m*-table. An *m*-table is associated with the central entity and one or more dimensions. Note that A_m can be an empty set. In such cases, the *m*-table just keeps the relationships among the central entity and the associated dimensions, and the records in the *m*-table may be used for counting occurrences instead of numeric aggregation.

Each summarisability constraint, $S_k \in S$, is a triple (D_i, M_j, Ω), where $D_i \in D$, $M_j \in M$, and Ω is an aggregation operator. In our biomedical data warehouse, the commonly used aggregation operators are {SUM, AVG, MAX, MIN, COUNT, CORRELATION, T-TEST, ANOVA}. The last three are statistical operators that are widely used in biomedical research. The CORRELATION operator is used to compute the Pearson or Spearman correlation coefficient between two random variables, T-TEST is used to determine if there is a significant difference between two random variables by computing the *t*-statistic and ANOVA (analysis of variance) is used to test whether there are differences between any pairs of random variables.

Summarisability constraints are critical for meaningful OLAP analyses of clinical and gene expression data. For example, it may not make sense to simply take the sum or average of microarray data for a group of genes (summarisation over the gene dimension). However, it is of interest to mine microarray data for gene relationships using the CORRELATION operator (pair-wise analysis of gene expression correlation). We are currently applying the traditional and statistical operators to biomedical data, and defining summarisability constraints for the OLAP operations.

Figure 5 shows the typical structure of a *BioStar* schema, which has one table for the central entity (CentralEntity), six dimension tables (Dimension1–6) and five *m*-tables (MTable1–5). Note that MTable5 is associated with two dimensions (Dimension5 and Dimension6). This is allowed by the *BioStar* model, in which an *m*-table may be associated with multiple dimensions. The *m*-tables often represent many-to-many relationships between the central entity and dimensions. Each *m*-table includes the primary key of the central entity table and the primary key(s) of the associated dimension table(s) as the foreign keys of the *m*-table. An *m*-table may contain zero, one, or more measures. In Figure 5, MTable3 does not have a measure attribute, but it is necessary for keeping the many-to-many relationship between CentralEntity and Dimension3; MTable4 has two measures (Measure 3 and Measure 4) associated with Dimension4; and the other *m*-tables have one measure attribute. Each *m*-table may also include non-measure attributes that are used to characterise the relationship between the central entity and dimension(s), or specify the temporal validity of the measure. The dimension tables in a *BioStar* schema may be denormalised as in a star schema, or normalised so that explicit dimension hierarchies can be defined.

Figure 5 Typical structure of a *BioStar* schema



4.2 Addressing the requirements

The *BioStar* model supports all the requirements described in Section 3.2. First, the *BioStar* model has the property of great extensibility, which is important for some fast-evolving data spaces such as the clinical and gene data spaces. In these data spaces, existing dimensions may need to be modified, and new dimensions may be added over time. *BioStar*'s extensibility is realised by storing different measures in separate *m*-tables.

In such a configuration, an existing dimension and its associated *m*-table can be modified independently from the other dimensions and *m*-tables. If a new dimension needs to be added, we can 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 central entity as one of the foreign keys to establish the relationship between the new dimension and the central entity. The representation of different measures using separate *m*-tables also reflects the natural situation of some complex biomedical data spaces, in which different dimensions are loosely coupled and each dimension often has its own measures.

Second, the many-to-many relationships between the central fact entity and dimensions are handled using the *m*-tables. For each of the many-to-many relationships, an *m*-table is created. This is similar to the treatment of many-to-many relationships in a standard Entity-Relationship schema.

Third, uncertain relationships between the central entity and dimensions may be kept in the *m*-tables. An additional field may be included in the *m*-table to specify if a relationship instance is uncertain. Furthermore, an *m*-table is normally small in size, when compared with the large central fact table of a star schema. It may be more efficient to update individual *m*-tables for uncertain relationships or imprecise data entries than the large fact table of a star schema.

Fourth, the *BioStar* model allows an *m*-table to have non-measure attributes that include single- or bi-temporal support for a measure. This feature is important for clinical data analysis, and is discussed further in the Section 4.3.

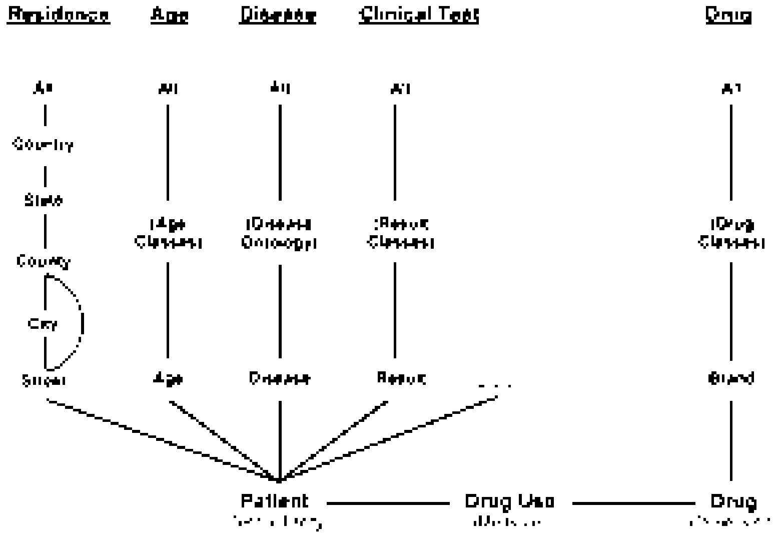
Finally, the *BioStar* model can be used to handle the commonly incomplete data from biomedical studies. If incomplete data are stored in the central fact table of a star schema, null values may need to be used for some missing measures and their associated dimension keys (foreign keys of the fact table). In a *BioStar* schema, since each *m*-table and its associated dimension table(s) can be populated independently from the other tables, incomplete data can be stored in the data warehouse by using only the relevant *m*-tables, and thus no null values are needed for the missing measures of some dimensions.

4.3 Support for OLAP

We now discuss *BioStar*'s properties for supporting OLAP operations. We assume that classification hierarchies are defined for the dimensions and the central entity. For example, gene ontology (GO) may be used to classify gene functions, and disease ontology may be used as the Disease dimension hierarchy in the clinical data space. These ontologies encode significant background knowledge of biology and thus are important for meaningful OLAP analyses of clinical and genomic data. The classification hierarchies of the other dimensions may also be defined in a similar way based on domain-specific knowledge.

For each measure in a *BioStar* schema, a data cube is pre-computed and applied to OLAP operations using the classification hierarchies of the dimension(s) and the central entity. While the dimension hierarchy is defined according to the characteristics of the dimension entity, the hierarchies associated with the central entity may be based on the other dimensions and measures in the data space. For example, the hierarchies shown in Figure 6 may be used for data cube construction and OLAP of the patient drug use measure. While the hierarchy of the Drug dimension is constructed based on the various drug classes, classification hierarchies of the central entity (Patient) may be defined according to the disease (the Disease hierarchy), demographics (the Residence and Age hierarchies), clinical test result (the Clinical Test hierarchy), and other dimensions. Note that, except for the Residence hierarchy, we do not show in Figure 6 the detailed structure of the hierarchies because some are very complex (e.g., disease ontology and drug classification) and the others may depend on application context (e.g., classes of age and clinical test result). The top element in the hierarchies (All) corresponds to the highest concept level. Numeric values of the clinical test result may be discretised and used to construct a classification hierarchy for Patient.

Figure 6 Classification hierarchies of Patient and Drug

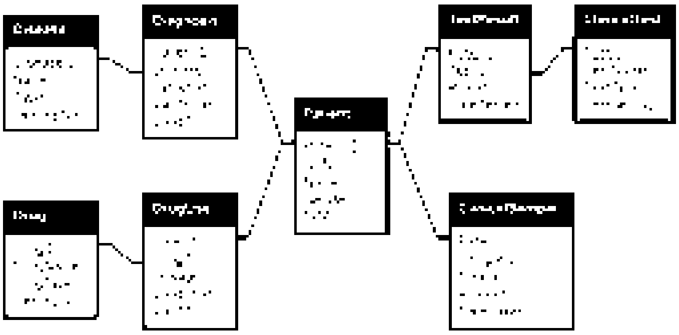


Thus, one novel feature of *BioStar* is that the central entity can have many classification hierarchies, some or all of which may be used for constructing any particular data cube and subsequently for OLAP operations. Conventional models may define alternative hierarchies for a given dimension, but they do not support the above feature. For example, if star schema is used to model the drug use example shown in Figure 6, the central fact table may need to be directly linked to all the dimensions that provide the classification hierarchies. This reduces model flexibility and results in redundancy in the fact tables, considering there are many measures in a single data space.

5 Case studies

We now use *BioStar* to design logical models for the clinical and gene data spaces. Figure 7 shows part of the *BioStar* schema for the clinical data space. The idea is to split the different clinical measures along the dimensions, which often results in separate *m*-tables. For example, ClinicalTest is a dimension table, and the clinical test results are kept in the *m*-table, TestResult. We include the primary keys of the central entity and dimension(s) in the corresponding *m*-table. Note that the one-to-many relationship between Patient and ClinicalSample allows the clinical sample measure to be included in the ClinicalSample table.

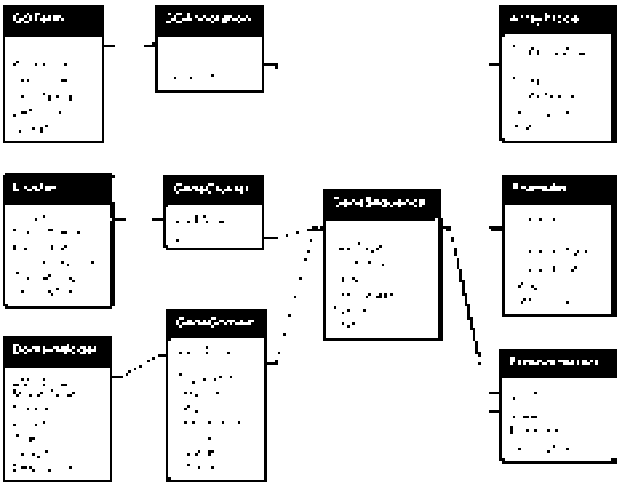
Figure 7 Part of the *BioStar* schema for the clinical data space



Two types of temporal support are provided for the different *m*-tables in Figure 7. One is the typical one-point support to indicate when the result is obtained or sample taken. The other is bi-temporal support, which specifies the time interval during which the measure is valid. For example, the Diagnosis measure table has ValidFrom and ValidTo to specify the valid time interval of a patient's disease.

Figure 8 shows part of the *BioStar* schema for the gene data space. GeneSequence is the central entity table, which has UID (unified gene identifier) as the primary key. Each of the *m*-tables (GOAnnotation, GeneCluster and GeneDomain) or dimension tables (ArrayProbe and Promoter) uses UID as a foreign key. There are probably no meaningful numeric measures for aggregation in this data space. The advantages of the *BioStar* schema, when compared with a standard Entity-Relationship schema, include high understandability to users and support for efficient navigation and selection of specific gene subsets and related information.

Figure 8 Part of the *BioStar* schema for the gene data space



An important consideration for the gene data model is extensibility. Due to the fast evolution of genomics, new dimensions may need to be added in the gene data space. For example, assume that we have implemented the *BioStar* schema shown in Figure 8, and populated the data warehouse. Now, we want to add a protein structure information dimension to the gene data space. We only need to add two additional tables, ProteinStructure and StructureSequence. ProteinStructure is the dimension table, which may use Protein Data Bank (PDB) identifiers as the primary key (PDBID) and store protein structure information such as compound name, structure resolution, etc. Since ProteinStructure has a many-to-many relationship with GeneSequence (i.e., one structure may consist of many chains, and one sequence may be related to many structure records), we add the StructureSequence *m*-table that uses UID and PDBID as foreign keys. These two tables for the new dimension can be populated without affecting the other tables in the data warehouse. In contrast, introduction of a new dimension into a conventional star schema would require re-computing of all the data entries in the fact table.

The snowflake schema shown in Figure 4 for the microarray data space may be viewed as a special case of *BioStar* schemas. If we select ClinicalSample or ArrayProbe as the central entity (depending on use cases), mRNAExpression is the *m*-table, which has the other three dimensions associated with it. Recall from the definition of *BioStar* schemas that an *m*-table can be associated with multiple dimensions.

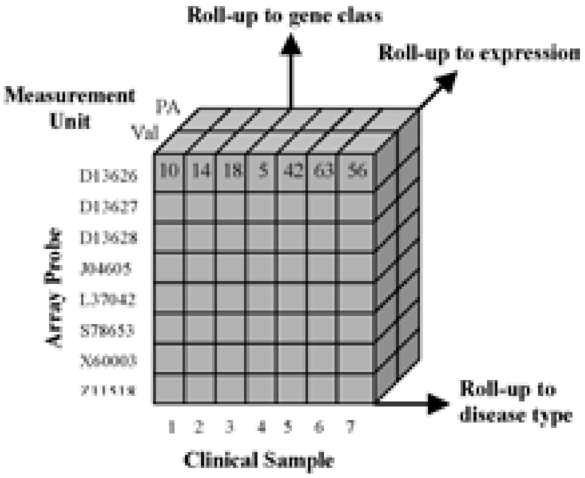
The above examples demonstrate that the *BioStar* model can be applied to a variety of biomedical data spaces. The high flexibility of *BioStar* schemas facilitates the integration of data from heterogeneous sources. To study complex biomedical problems, it is essential to combine data from many different sources. In the remainder of this section, we give a case study for using the multidimensional data model to integrate disease information with microarray gene expression data. OLAP operations are then used to identify informative genes of tumours.

Ramaswamy et al. (2001) used Affymetrix GeneChips oligonucleotide microarrays (representing 16,063 genes) to survey gene expression in 218 tumour samples of 14 common tumour types and 90 normal tissue samples. The dataset contains about 5 million gene expression values, and is available at http://www-genome.wi.mit.edu/MPR/GCM.html. The authors used the dataset for classification of tumours and identification of tumour marker genes.

The gene expression dataset can be viewed as a data cube with three dimensions: ClinicalSample, ArrayProbe and MeasurementUnit (Figure 9). The Experiment dimension is not considered in this case study because all the gene expression values are processed in the same way. The ClinicalSample dimension captures information about the 308 tumor and normal tissue samples. The dimension hierarchy may be defined as a lattice consisting of four concept levels: individual ‘patient’ level; specific ‘tumour_type’ or ‘normal_tissue_type’ level; ‘all_tumor’ or ‘all_normal_tissue’ level; and ‘all_sample’ level. The ArrayProbe dimension relates oligonucleotide probes to gene sequences. A gene is often represented by a set of probes in a microarray. For the MeasurementUnit dimension, the Affymetrix GeneChip platform provides two kinds of gene expression measurements for each probe set: a presence/absence (PA) call and a numeric value (average gene expression level). The PA call is based on statistical analysis of hybridisation signals over the probe set and can take one of the following three labels: ‘P’ for ‘present’, ‘M’ for ‘marginal’ and ‘A’ for ‘absent’ of gene expression.

Now, we are set to identify genes whose expression is significantly changed (up-regulated or down-regulated) in tumours. First, we summarise the gene expression data over the MeasurementUnit dimension. There are several ways (aggregation functions) to do the summarisation. For this study, we set a gene expression value to zero if the PA call is ‘A’, and leave the other values unchanged (if the PA call is ‘P’ or ‘M’).

Figure 9 OLAP for microarray data exploration



Second, we roll-up the gene expression data over the ClinicalSample dimension from ‘patient’ level to the next higher level, which is either ‘tumor_type’ for tumour samples or ‘normal_tissue_type’ for normal tissue samples. The aggregation functions used in this operation compute the mean and variance of expression values for each gene. After this summarisation, each cell in the data cube contains three values: mean, variance and the number of values aggregated.

Third, the slice of a particular tumour type and that of the corresponding normal tissue type are selected. Student’s t test is then applied to each pair of selected cells for each gene. The statistical test calculates a p-value for each gene. Because of the multiple comparisons in the statistical test, the p-values are adjusted using the Bonferroni correction method.

Finally, over the ArrayProbe dimension, we select genes that have p-values less than a threshold ($p < 0.05$ was used in this study). At the 5% significance level, we discovered many genes whose expression is changed in tumours when compared with corresponding normal tissues. For example, comparison of eleven renal carcinoma tumour samples with thirteen normal kidney samples revealed that an insulin-like growth factor 2 gene (GenBank accession: M17863) is significantly over-expressed in renal carcinoma ($p = 0.033$). In another comparison of eleven pancreas adenocarcinoma tumour samples with ten normal pancreas samples, a guanine nucleotide-binding protein alpha-subunit gene (M21142, $p = 0.002$) and an EST cloned from parathyroid tumours (W55861, $p = 0.047$) were found to be among the informative genes for pancreas adenocarcinoma.

The above procedure can be designed to allow users to guide the exploration in the gene expression data space. For example, a user may roll-up the gene expression data over the ClinicalSample dimension up to ‘all_tumor’ and ‘all_normal_tissue’ level to discover genes whose expression is significantly changed in all the tumours. The user may also choose different aggregation operators and parameters during the exploration. With the rapid accumulation of clinical and genomic data, we believe that such a multidimensional data warehouse with interactive OLAP tools will be important components of an integrated platform for biomedical data analysis.

6 Conclusions

We described a new multidimensional data model called *BioStar* for clinical and genomic data. *BioStar* schemas are able to capture the complex data structures and semantics. The model has the properties of great extensibility and flexibility to be widely applicable to biomedical data. *BioStar*'s extensibility and flexibility are realised by storing different measures in separate *m*-tables. These *m*-tables are used to handle the many-to-many relationships between the central entity and dimensions and can be designed to support specific features of a measure (e.g., bi-temporal support for some clinical data). Since each *m*-table and its associated dimension table(s) can be populated independently from other *m*-tables, incomplete data can be stored in the data warehouse by using only the relevant *m*-tables. In addition, it is more efficient to update an *m*-table for uncertain relationships or imprecise data entries than the large central fact table of a star schema. In the future, we will investigate other issues for application of data warehousing and OLAP technologies to biomedical research, including query optimisation, definition of classification hierarchies and modelling the data warehouse construction process.

Acknowledgements

We thank Xian Xu and Daxin Jiang for helpful discussions. This work was supported by a grant from the National Science Foundation.

References

Abello, A., Samos, J. and Saltor, F. (2001) 'A framework for the classification and description of multidimensional data models', *12th International Conference on Database and Expert Systems Applications (DEXA), volume 2113 of LNCS*, pp.668–677.

Cabibbo, L. and Torlone, R. (1998) 'Querying multidimensional databases', *Proceedings of the 6th International Workshop on Database Programming Languages*, pp.319–335.

Chen, J., Zhao, P., Massaro, D., Clerch, L.B., Almon, R.R., DuBois, D.C., Jusko, W.J. and Hoffman, E.P. (2004) 'The PEPR GeneChip data warehouse, and implementation of a dynamic time series query tool (SGQT) with graphical interface', *Nucl. Acids. Res.*, Vol. 32, pp.D578–D581.

Cornell, M., Paton, N.W., Wu, S., Goble, C.A., Miller, C.J., Kirby, P., Eilbeck, K., Brass, A., Hayes, A. and Oliver, S.G. (2001) 'GIMS – a data warehouse for storage and analysis of genome sequence and functional data', *Proceedings of the IEEE 2nd International Symposium on Bioinformatics and Bioengineering (BIBE)*, pp.15–22.

Datta, A. and Thomas, H. (1999) 'A conceptual model and algebra for on-line analytical processing in data warehouses', *Decision Support Systems*, Vol. 27, No. 3, pp.289–301.

Dubitzky, W., Krebs, O. and Eils, R. (2001) 'Minding, OLAPing, and mining biological data: towards a data warehousing concept in biology', *Proceedings of Network Tools and Applications in Biology (NETTAB), CORBA and XML: Towards a Bioinformatics Integrated Network Environment*, pp.78–82.

Gollub, J., Ball, C.A., Binkley, G., Demeter, J., Finkelstein, D.B., Hebert, J.M., Hernandez-Boussard, T., Jin, H., Kaloper, M., Matese, J.C., Schroeder, M., Brown, P.O., Botstein, D. and Sherlock G. (2003) 'The Stanford Microarray database: data access and quality assessment tools', *Nucl. Acids. Res.*, Vol. 31, pp.94–96.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gassenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, D.D. and Lander, E. S. (1999) 'Molecular classification of cancer: class discovery and class prediction by gene expression monitoring', *Science*, Vol. 286, No. 15, pp.531–537.

Inmon, W.H. (1996) *Building the Data Warehouse*, 2nd ed., John Wiley & Sons, New York.

Kimball, R. (1996) *The Data Warehouse Toolkit*, John Wiley & Sons, New York.

Marcel, P. (1999) 'Modeling and querying multidimensional databases: an overview', *Networking and Information Systems Journal*, Vol. 2, pp.515–548.

Markowitz, V.M. and Topaloglou, T. (2001) 'Applying data warehouse concepts to gene expression data management', *Proceedings of the IEEE 2nd International Symposium on Bioinformatics and Bioengineering (BIBE)*, pp.65–72.

Paton, N.W., Khan, S.A., Hayes, A., Moussouni, F., Brass, A., Eilbeck, K., Goble, C.A., Hubbard, S.J. and Oliver, S.G. (2000) 'Conceptual modeling of genomic information', *Bioinformatics*, Vol. 16, No. 6, pp.548–557.

Pedersen, T.B., Jensen, C.S. and Dyreson, C.E. (2001) 'A foundation for capturing and querying complex multidimensional data', *Information Systems*, Vol. 26, No. 5, pp.383–423.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J.P., Poggio, T., Gerald, W., Loda, M., Lander, E.S. and Golub, T.R. (2001) 'Multiclass cancer diagnosis using tumor gene expression signatures', *Proc. Natl. Acad. Sci.*, USA, Vol. 98, No. 26, pp.15149–15154.

Vassiliadis, P. (1998) 'Modeling multidimensional databases, cubes and cube operations', *Proceedings of the 10th International Conference on Scientific and Statistical Database Management*, pp.53–62.

Vassiliadis, P. and Sellis, T. (1999) 'A survey of logical models for OLAP databases', *ACM SIGMOD Record*, Vol. 28, No. 4, pp.64–69.