

Chapter 2

Bioinformatics for Transporter Pharmacogenomics and Systems Biology: Data Integration and Modeling with UML

Qing Yan, M.D., Ph.D.

Abstract

Bioinformatics is the rational study at an abstract level that can influence the way we understand biomedical facts and the way we apply the biomedical knowledge. Bioinformatics is facing challenges in helping with finding the relationships between genetic structures and functions, analyzing genotype–phenotype associations, and understanding gene–environment interactions at the systems level. One of the most important issues in bioinformatics is data integration. The data integration methods introduced here can be used to organize and integrate both public and in-house data. With the volume of data and the high complexity, computational decision support is essential for integrative transporter studies in pharmacogenomics, nutrigenomics, epigenetics, and systems biology. For the development of such a decision support system, object-oriented (OO) models can be constructed using the Unified Modeling Language (UML). A methodology is developed to build biomedical models at different system levels and construct corresponding UML diagrams, including use case diagrams, class diagrams, and sequence diagrams. By OO modeling using UML, the problems of transporter pharmacogenomics and systems biology can be approached from different angles with a more complete view, which may greatly enhance the efforts in effective drug discovery and development. Bioinformatics resources of membrane transporters and general bioinformatics databases and tools that are frequently used in transporter studies are also collected here. An informatics decision support system based on the models presented here is available at <http://www.pharmtao.com/transporter>. The methodology developed here can also be used for other biomedical fields.

Key words: Bioinformatics, pharmacogenomics, systems biology, data modeling, data integration, object oriented, Unified Modeling Language, computational, decision support, transporters, drug development, databases.

1. Bioinformatics and Membrane Transporter Studies

With a history of less than 40 years, bioinformatics is a rapidly growing area that applies computational approaches to solve biological problems. Similar to the composition of the word itself, “bioinformatics” is an independent field developed from the union of computer science and molecular biology. Rather than a simple combination of computer science and biology, bioinformatics should be an “organic” integration of the two. This merge was started in 1970s, when it was found that RNA secondary structure might be predicted with computational techniques (1, 2). At that time, people began to build databases of nucleic acids (3) and proteins (4). Algorithms and programs were developed to translate DNA sequences into protein sequences (5, 6) and to detect patterns including restriction enzyme recognition sites (7, 8).

Various bioinformatics approaches have been used in transporter studies. For example, the sequence similarity searching tool BLAST (Basic Local Alignment Search Tool) has been used extensively in the analysis of transport systems in different organisms (9) and in the identification of transporter genes (10). The database PROSITE has been used for functional analysis in transporter genes (11, 12).

Table 2.1 lists some bioinformatics resources designed specifically for membrane transporter studies (Websites accessed in May 2009). **Table 2.2** summarizes some general bioinformatics databases and tools that are frequently used in

Table 2.1
Data sources for membrane transporter and ion channel studies

Category	Databases and tools	Links
Transporter Portal	Human Membrane Transporter Database Portal	http://www.pharmtao.com/transporter
Transporter Classification	Transport Classification Database (TCDB)	http://tcdb.ucsd.edu/index.php
Genomic Comparisons	TransportDB	http://www.membranetransport.org/
Membrane Proteins in Different Species	Human membrane protein library (HMPL)	http://wardlab.cbs.umn.edu/human/
	Functional Genomics of Plant Transporters (PlantsT)	http://plantst.genomics.purdue.edu/

(continued)

Table 2.1 (continued)

Category	Databases and tools	Links
	Aramemnon : Plant membrane protein database	http://aramemnon.botanik.uni-koeln.de/
	Arabidopsis Membrane Protein Library	http://wardlab.cbs.umn.edu/arabidopsis/
	Rice Membrane Protein Library (RMPL)	http://wardlab.cbs.umn.edu/rice/
	Yeast membrane protein library (YMPL)	http://wardlab.cbs.umn.edu/yeast/
	<i>Schizosaccharomyces pombe</i> membrane protein library (SpMPL)	http://wardlab.cbs.umn.edu/pombe/
	Drosophila membrane protein library (DMPL)	http://wardlab.cbs.umn.edu/fly/
	<i>C. elegans</i> membrane protein library (CeMPL)	http://wardlab.cbs.umn.edu/worm/
Ion Channels	Ligand-Gated Ion Channel database	http://www.ebi.ac.uk/compneur-srv/LGICdb/LGICdb.php
	Ion Channel Diseases	http://neuromuscular.wustl.edu/mother/chan.html
	Voltage-gated potassium channel database (VKCDB)	http://vkcdb.biology.ualberta.ca/
	ChannelDB	http://www.modelersworkspace.org/channeldb/ChannelDB.html
ABC Transporters	ABCISSE database	http://www1.pasteur.fr/recherche/unites/pmtg/abc/database.iphtml
	Human ATP-Binding Cassette Transporters	http://nutrigene.4t.com/humanabc.htm
	ABC Transporter Genes Database	http://www.humanabc.bio.titech.ac.jp/
	P-type ATPases database	http://biobase.dk/~axe/Patbase.html
	Arabidopsis ABC superfamily	http://www.arabidopsis.org/info/genefamily/ABC_proteins.html
	Archaeal and Bacterial ABC transporter database	http://www-abcdb.biotoul.fr/
Specific Diseases	Wilson Disease Mutation Database	http://www.wilsondisease.med.ualberta.ca/database.asp

Table 2.2
General bioinformatics databases and tools for membrane transporter studies

Category	Database/tool example	URL
Nucleotide and Protein Portal	Entrez	http://www.ncbi.nlm.nih.gov/Entrez/
	European Bioinformatics Institute (EBI)	http://www.ebi.ac.uk/
Homology search	BLAST	http://www.ncbi.nlm.nih.gov/BLAST/
Multiple alignment	Clustal W	http://www.ebi.ac.uk/clustalw/
DNA	dbEST	http://www.ncbi.nlm.nih.gov/dbEST/index.html
Gene-oriented cluster	UniGene	http://www.ncbi.nlm.nih.gov/UniGene/index.html
Sequence variation	dbSNP	http://www.ncbi.nlm.nih.gov/SNP/index.html
	International HapMap Project	http://www.hapmap.org/
Human Gene Mutation Database	HGMD	http://www.uwcm.ac.uk/uwcm/mg/hgmd0.html
Motif analysis	Pfam	http://pfam.sanger.ac.uk/
	ProfileScan	http://hits.isb-sib.ch/cgi-bin/PFSCAN
Exon finding and gene annotation	GenScan	http://genes.mit.edu/GENSCAN.html
Secondary structure prediction	PredicProtein	http://cubic.bioc.columbia.edu/predictprotein/submit_def.html
Transmembrane region detection	TMPred	http://www.ch.embnet.org/software/TMPRED_form.html
Structure (3D) database	PDB	http://www.rcsb.org/pdb/
3D structure prediction	Geno3D	http://geno3d-pbil.ibcp.fr/
Pathway and cellular regulation	KEGG	http://www.genome.ad.jp/kegg/kegg2.html
	Reactome	http://www.reactome.org/
	Human Protein Reference Database (HPRD)	http://www.hprd.org/
Disorders	Pathguide	http://www.pathguide.org/
	OMIM	http://www.ncbi.nlm.nih.gov/omim/
	Genes and Disease Map	http://www.ncbi.nlm.nih.gov/disease/Transporters.html
Literature	PubMed	http://www.ncbi.nlm.nih.gov/PubMed/

transporter studies (Websites accessed in May 2009). These bioinformatics databases and tools are linked and integrated in a comprehensive database portal for transporters at <http://www.pharmtao.com/transporter> (accessed in May 2009).

As we enter the transition era from structural to functional genomics and proteomics, especially with the overwhelming variety and volume of data, bioinformatics becomes increasingly important and indispensable for other biomedical sciences. Different from most traditional biomedical sciences that are grounded in the observation of the physical world, bioinformatics is the rational study at an abstract level that can influence the way we understand biomedical facts and the way we apply the biomedical knowledge. At this stage, bioinformatics is facing challenges in helping with finding the relationships between genetic structures and functions, analyzing genotype–phenotype associations, and understanding gene–environment interactions at the systems level.

One of the most important issues in bioinformatics is data integration. This includes the integration of data from heterogeneous resources, from various data types, and enterprise-wide data integration among different groups and departments. There can be valuable knowledge buried in various unorganized data, and the process of data integration can help “unveil” the hidden knowledge. This chapter briefly introduces methodologies on how to extract useful information from various data so that the information can be applied directly in research and development projects.

With the volume of data and the high complexity, computational decision support is essential for integrative transporter studies in pharmacogenomics, nutrigenomics, epigenetics, and systems biology. For the development of such a decision support system, object-oriented (OO) models can be constructed using the Unified Modeling Language (UML). The modeling methods for decision support in transporter studies of these emerging fields will be described in detail, such as how to construct UML diagrams based on biomedical models at different system levels. The methodology developed here can also be used for other biomedical fields.

2. Data Integration Methods in Membrane Transporter Studies

Data integration is not only just for simple data access but also for knowledge discovery and decision support. The two words “data” and “information” are often used interchangeably. In fact, they are quite different. The term “data” implies a collection of

discrete elements, such as a file. Data are rarely clean and may have different formats. Some data are from multiple competing sources. Some data have missing and incomplete fields. In addition, data formats and contents may change over time. When data are cleaned, structured, merged, aggregated, derived, sorted, and displayed, they become “information.”

Usually a database system provides an area to collect, integrate, and store data to perform the actions to enrich and enhance the value of the data. A database system offers a platform to transform data into information and is often useful for decision support purposes. The data integration methods introduced here can be used when researchers try to organize and integrate public and in-house data. This kind of work has become crucial in the routine lab work, in order to organize and even publish one’s research results.

Figure. 2.1 shows the data transformation and integration process. This is also a process that standardizes names and values, resolves inconsistencies in representation of data, and integrates common values together. The “equal” values of data from disparate sources that represent the same biomedical facts are also resolved in this process. This transformation process is repeated over and over again, during the original development of the target database, when adding new sources to the existing database, and when distributing data from the system to users.

Here we focus on introducing the data consolidation approach in data integration, although other approaches can be used, such as the method of federation (13). The consolidation approach is based on constructing a database with a single large data model, e.g., when we need to extract data from various data sources and centralize these extracted data at one place. The major

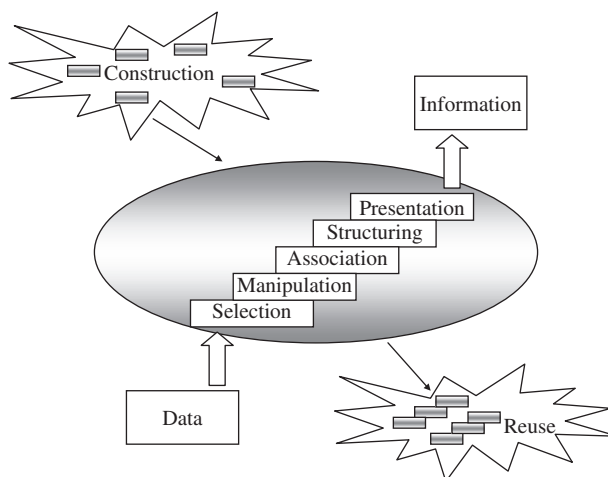


Fig. 2.1. Data integration: the process from data to information.

benefits of this approach include enforcement of the standardization of heterogeneous data.

The data integration and transformation process begins with the selection of data sources, as shown in **Fig. 2.1**. Data are selected through the screening of all available sources and choosing the ones that can best fulfill the requirements. For transporter studies, data sources can be from the tools and databases listed in **Tables 2.1** and **2.2**. The selected data are then manipulated and transformed. Data consolidation is a procedure that analyzes and merges data from disparate sources or systems into a single, integrated data structure. This is achieved through identifying data that is common across the various source files, and investigating the rules that manage the usage of the data. For example, polymorphism data about transporter genes can be retrieved from several sources, such as databases dbSNP and OMIM (Online Mendelian Inheritance in Man). It is necessary to integrate data from these different sources into a common data structure, which includes variation types such as deletion or point mutation. Such data structure can be constructed based on the data models developed during the requirement analysis phase, which will be introduced in detail in the next section (*see Section 3.2*).

The process of data consolidation also includes identifying data elements that have common biomedical meaning even though the names are different (synonyms) or those that have the same name but represent different biomedical facts (homonyms). Failing to properly identify these in the source files may result in disparate data that fail to provide the true integration points. For example, failure to identify the synonyms of one transporter gene may cause these different names to be regarded as different genes, which may lead to unclear data and serious data redundancy, even repeated experiments. The classification of transporter genes can provide a comprehensive view of all transporters and help elucidate the directions of transporter research (14). Transporter gene classification is also one of the most important parts involved in data consolidation.

During the data manipulation process, data are also cleaned. In this step, redundant data are removed, and outdated data are updated. Data cleaning can be done together with data consolidation and conversion. For example, sometimes the synonyms of a transporter gene are recorded as different records with redundant gene sequences. These redundancies can be removed with the identification of the synonyms. When the data are manipulated through consolidation, conversion, and cleaning, the result can be presented with a structuring of the information. Such data presentation may include tables and graphs, which can also be used in the decision support process.

3. Data Modeling for Informatics Support of Transporter Phar- macogenomics and Systems Biology

3.1. Informatics Support in Pharma- cogenomics and Systems Biology

Pharmacogenomics is multi-disciplinary involving molecular biology and human genetics, genomics, bioinformatics, physiology, pharmacology, and internal medicine (13, 15). As discussed in **Chapter 1**, the emerging fields of nutrigenomics, epigenetics, and systems biology add more dimensions to this complexity. It is even difficult for experts from these different domains to communicate with each other. These multi-level characteristics and domain knowledge barriers bring great challenges to decision making in clinics and labs. Comprehensive and integrative informatics methodologies are needed to break these barriers and to improve the information flow for better communication.

For example, genetic variations have been suggested to be useful for decision making about drug treatment in clinics (16). To achieve this goal effectively, information of genetic variations needs to be processed and provided by a computational system to support the clinical decision making. In addition, the application of high-throughput technologies and the analysis of patient genetic profiles require powerful informatics support. With the amount of available data rapidly increasing, the need for strong information technology support becomes increasingly urgent.

A bioinformatics decision support system (DSS) is a system that provides information to assist biomedical experts in making decisions and doing their job more effectively in both laboratory research and clinical practice. Such systems can help record, store, analyze, and mine the data. Here a decision is an irreversible choice among alternative ways to allocate valuable resources.

To build such informatics systems, intercommunication and interoperation between different biomedical databases are becoming critical issues. To solve these problems, bioinformatics is demanding a common literacy with mutual intelligibility that can be widely accepted (17).

This goal has been difficult to achieve and has been considered to be the first obstacle in biological knowledge modeling and encoding (18). These problems are central to providing informatics support for transporter pharmacogenomics and systems biology studies because this is a complex area requiring heterogeneous data sources. In addition, the special features of phar-

macogenomics, such as genetic polymorphisms and genotype–phenotype correlations, are difficult to process using the traditional passive and static flat files or even relational models.

To solve these problems and construct useful informatics support systems, advanced and integrative data models, such as using the object-oriented (OO) methodology, need to be built. A data model covers the scope of the system development including relationships, attributes, and definitions. Data modeling provides a formal methodology for documenting users' data needs. The models we use for building pharmacogenomics and systems biology computational systems will have a profound influence upon how a scientific or clinical problem is attacked, how a solution is shaped, and how a result is interpreted.

3.2. Unified Modeling Language (UML)

To construct an accurate and usable model for transporter pharmacogenomics and systems biology, it is necessary to capture the important concepts and relationships of the domain knowledge, and convert such understanding into physical data structures for decision support systems (*see Section 2*). Building a biomedical model from the domain requirement analysis is necessary for approaching the complexity. This model is scientific in nature and consists of accurate description and illustration of the fundamental factors and processes of our understanding of this particular area of science.

From this scientific model, important and repetitively occurring concepts and factors can be abstracted and identified. These concepts and factors can be viewed as objects in the sense of object-oriented (OO) methodology. In real-world terms, an object can be defined as a concept, abstraction, or a thing with crisp boundaries and meanings for the problem at hand (19). In software terms, an object is an intelligent piece of a program that can encapsulate code and data. These objects, together with the interrelationships among them, should be able to represent the outputs as well as the inner workings of the biomedical model. These objects are also our building blocks for constructing sophisticated information systems.

Based on the biomedical model and abstracted concepts, OO models can be constructed using the Unified Modeling Language (UML). Object-oriented methods offer a unifying paradigm for the three traditional phases of software development: analysis, design, and implementation (20). This unification leads to a smooth transition from one phase to the next. UML is an object-oriented design language for specifying, visualizing, constructing, and documenting the objects of a system (21). It is a standard modeling language that has been widely accepted in computer science and used extensively in the business world. The application of UML can help the transformation from the logical model to the physical mode smoothly.

In recent years, UML has been adopted by more and more biomedical systems, especially in the medical imaging field (22). It was used in the implementation of brain computer interface (BCI) systems (23). In the Biomedical Research Integrated Domain Group (BRIDG) project, declarative and procedural knowledge were represented with the UML class, activity, and state diagrams (24).

UML has been suggested as a useful language for cell and biochemistry modeling (25). The UML approach was adopted in systematic modeling, capturing, and disseminating proteomics experimental data (26). It has been used in the integration of microarray gene expression, proteomics, and metabolomics data in the Chemical Effects in Biological Systems (CEBS) through building the Systems Biology Object Model (SysBio-OM) (27).

To overcome the barriers between different knowledge domains and capture the essence of different disciplines for a coherent decision support system in pharmacogenomics and systems biology, a methodology for model construction can be used (13). This approach is from domain requirement analysis and biomedical models \Rightarrow concept abstraction and object design \Rightarrow OO UML models. The application of UML in this approach helps decompose the complexity and make the system comprehensible for data analysis in pharmacogenomics and systems biology. A DSS built based on this approach can represent the most important correlations and key issues in transporter studies discussed in **Chapter 1**. Information from the requirement analysis and the biomedical model will feed the data modeling directly.

To build the DSS for transporter pharmacogenomics and systems biology studies, the first phase is system design, which starts from understanding the biomedical needs and data requirements of the system users. Analyzing and designing are critical for the success of the system construction. The analysis process identifies what the problem is and what a system needs to do. The design process provides a logical solution so that the system satisfies the requirements. Once the requirements are specified, UML diagrams can be used to analyze, design, and develop applications. The following sections focus on this design phase.

The phase after the design is system implementation (13). A physical database system can be developed according to the data model designed in the previous phase. During this phase, data analysis and integration are performed to determine the best and cleanest source of data (*see Section 2*). The step after the implementation is system application. In this phase, data access tools are used to build reports and support data mining.

3.2.1. Biomedical Models at the Molecular and Systems Level

The biomedical system we are dealing with is overwhelmingly complex. It is necessary to decompose it into understandable chunks to comprehend and manage the complexity. To do this,

models can be constructed to describe, abstract, and represent essential aspects of the system. As mentioned earlier, the first step for model construction in pharmacogenomics and systems biology is domain requirement analysis and building biomedical models. The systematic overview of the knowledge domain based on biomedical models is necessary to clearly identify the target domain first. A biomedical model describes the natural scenarios and reflects the understanding and thinking process of domain users for whom the software system is built. It helps capture the most important issues that the users are concerned about. Biomedical modeling before data modeling can help lay the ground for further concept identification.

Biomedical models represent the problems in the biomedical system in an intuitive way using the language of the field. The requirement of a biomedical model includes that it should show the biological objects, associations, and processes as they are understood by biomedical experts. These models then need to be translated into data models that are the foundations of further software development. Based on these biomedical models, use cases can be described and concepts can be abstracted (*see Section 3.2.2*). The step of concept abstraction can lead to the data modeling phase with the creation of series of diagrams.

Figure 2.2 shows a biomedical model describing the structure–function, gene–drug, and genotype–phenotype correlations of transporters at the molecular level. Two typical transporters are used as examples in the model to illustrate different aspects. G1 is used to represent common transporter families such as those in the ABC superfamily, for example, multi-drug-resistance protein (MRP). MRPs are organic anion transporters that transport anionic drugs such as methotrexate, and neural drugs conjugated to acidic ligands such as sulfate (28, 29). Compounds can be transported by MRPs in complexes with glutathione (GSH). G2 is used to represent other types such as the families of ion channels. For example, intermediate conductance Ca^{2+} -activated K^{+} channel (IKCa1) modulates calcium influx by regulating the membrane potential and the driving force for calcium entry. These two kinds of transporters are also used to represent the correlations at two levels, i.e., protein and nucleotide levels.

Here the modeling part of G1 is used to focus on the description of protein structure and functions. The topologies of these genes include transmembrane domains (TMDs). More detailed topology of transporter proteins is also described, such as the nucleotide-binding domain (NBD), and a “signature” motif that defines the NBDs of ABC transporters (30).

Elevated levels of G1 can confer resistance to drugs. For example, overexpression of MRP2 was found to result in resistance to cisplatin, etoposide, doxorubicin, and epirubicin (31).

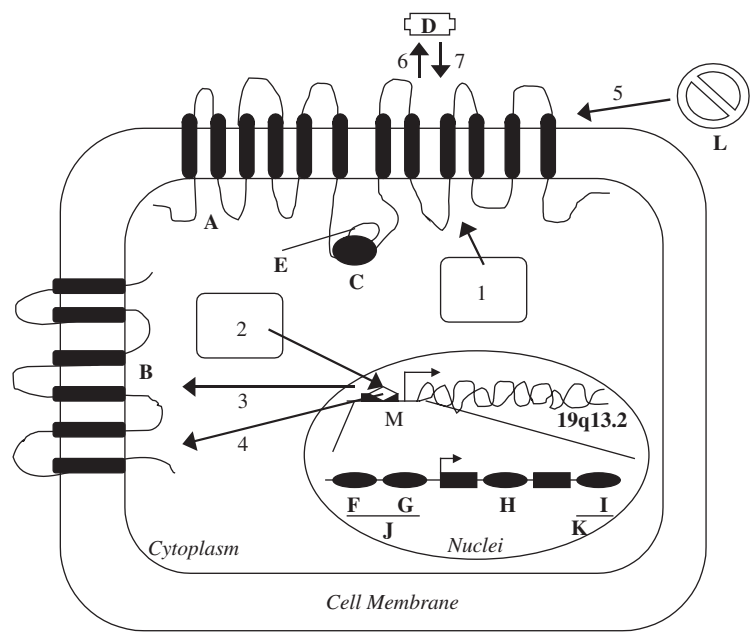


Fig. 2.2. The biomedical model of correlations of transporters at the molecular level. A: G1; B: G2; C: nucleotide-binding domain (NBD); D: Drug; E: "Signal" motif; F: Enhancer; G: Promoter; H: Intron; I: Silencer; J: 5'-region; K: 3'-region; L: Inhibitors; M: Regulatory unit. 1: Biochemical pathway; 2: PKC signaling pathway; 3: increase expression; 4: mutation activates expression; 5: inhibit transporter; 6: drug efflux; 7: drug influx.

Because of the potential involvement of these drug pumps in the clinical phenotypes such as drug resistance, inhibitors are also important in describing transporters' functions. For example, high-affinity substrates can be potent competitive inhibitors, such as leukotriene C4 and S-decylglutathione for MRP1 (32).

In the modeling part of G2, the structure–function correlation at the nucleotide level is emphasized to represent the common mechanisms in human genes. In the genome, IKCa1 is located at chromosome 19q13.2. IKCa1 can be upregulated through the stimulation of PKC pathway (e.g., in T cells), which can trigger transcriptional activation of the IKCa1 promoter. The regulatory regions of a gene include enhancer, promoter, and silencer. These regulatory units are located in 5'- and 3'-gene flanking regions and in introns. The locations of these regulatory elements and the nucleotide sequences can describe their structure features. Their corresponding functional characteristics can be described in the effect on gene transcriptional activity and tissue and stage specificities.

The major correlations, especially gene—drug and genotype–phenotype interactions, are described at the systems level in Fig. 2.3. Abnormal function of transporter proteins can cause abnormal phenotypes, i.e., diseases. Transporters can also be

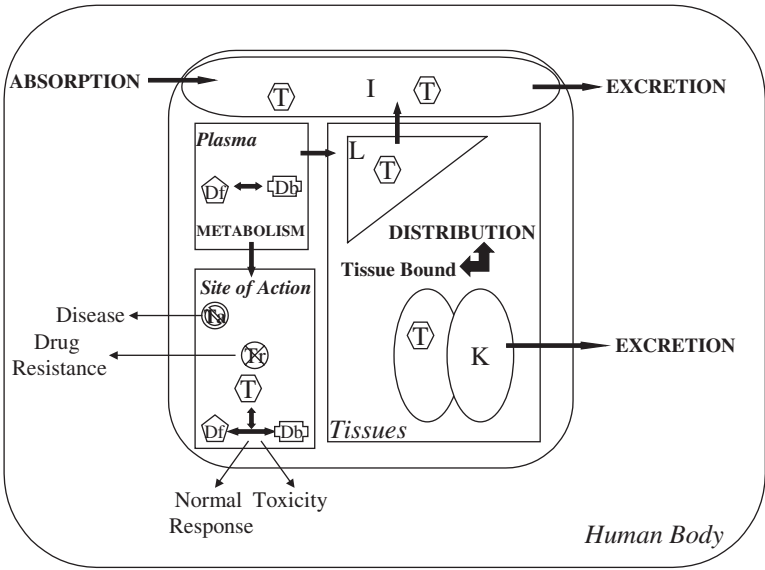


Fig. 2.3. The biomedical model of correlations of transporters at the systems level. *I*: Intestines; *L*: Liver; *K*: Kidney; *Df*: Free drug; *Db*: Bound drug; *T*: transporter; *Ta*: Altered transporters (that cause diseases); *Tr*: Transporters that are responsible for drug resistance.

involved in drug-response phenotypes of resistance, toxicity, or normal responses. The diagram illustrates the processes involved in drug transport and the effect of transporter actions on the bioavailability of drugs. Drug availability can be controlled by drug absorption and excretion, as shown in the diagram. Besides absorption and excretion, the interaction processes between drugs and the human body also include the distribution and metabolism. In addition, transporters are distributed in different tissues.

3.2.2. Use Case Diagrams

In UML terminology, user requirements are expressed in terms of *use cases*. A use case is a process that fulfills certain requirements of a system user (33). A process describes a sequence of events, actions, and transactions needed to complete something of usefulness to a user, from start to finish. Use case diagrams describe the main processes in a system and the interactions between the processes (use cases) and the external systems or actors. An actor is an entity outside the system that in some way participates in the story of the use case. Actors are represented by the role they play in the use case, such as a pharmacologist or a bioinformatician. A use case diagram defines the system boundaries, as well as the users that will utilize the system. Use cases comprise all the system functions identified during the prior requirement analysis and biomedical modeling. The processes in these use cases can be simple data query and retrieving, as well as more complicated

knowledge discovery. In the later case, the system is also functioning as a data mining tool.

In a use case diagram, a rectangle with rounded corners represents the application or system (*see Fig. 2.4*). Use cases are shown in ovals, with the name of the use case written inside the oval. Actors are represented in stick figures, with the name written under the figure. A line links the actors and the use cases they interact with.

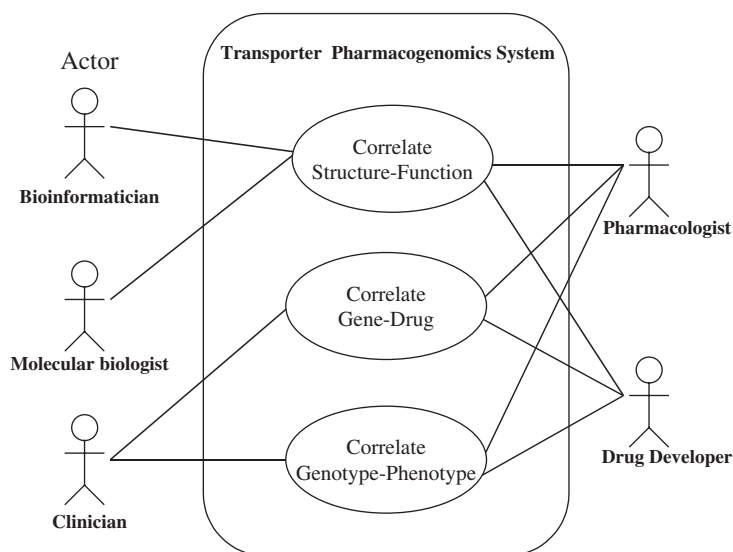


Fig. 2.4. The use case diagram of the transporter pharmacogenomics decision support system.

Figure. 2.4 shows the use case diagram of the transporter pharmacogenomics and systems biology decision support system (34). The actors in the diagram include molecular biologists, bioinformatician, pharmacologists, drug developers, and clinicians. The use cases that the system supports include, but not limited to, “Correlate Structure–Function,” “Correlate Gene–Drug,” and “Correlate Genotype–Phenotype.”

Pharmacologists and drug developers (the actors) may be interested in finding the gene–drug interactions. For example, genetic alterations in transporter genes BCRP and MXR are shown to be associated with resistance to mitoxantrone in breast cancer cell lines (35). It may be interesting to determine if other transporter genes are involved in the resistance. The actors can also categorize the genes and drugs with known interactions, which might help predict new interactions. For instance, to answer the question “For a new drug, what genes may interact with it?” analysis of the interaction patterns in drugs with similar structures and functions might be helpful. In this data mining

process, the information of the structure–function correlation is also important.

The study of genotype—phenotype correlation may be helpful to pharmacologists and drug developers to get some feedback about the use of drugs. This correlation information can assist more accurate drug targeting in the drug design process. For example, the identification of potential gene markers in the drug-resistance phenotype may provide clues for these actors to design new drugs targeting the markers to reverse or overcome the resistance. The software system will be a very useful tool in these decision-making processes (34).

3.2.3. Concept Abstraction and Class Diagrams

In UML, a class describes a set of objects that share the same attributes, methods, and relationships (33). A class diagram illustrates classes and the relationships between classes. In a class diagram (*see Fig. 2.5*), a large rectangle is divided into three horizontal compartments. The name of the class is written in the top section. The second section records the attributes of the class. The lower section usually includes operations and methods. The associations needed to record relationships are also added. An association is a relationship between objects that designate some meaningful and interesting connection. Associated classes are linked by lines. In some cases an association can be an object that has its

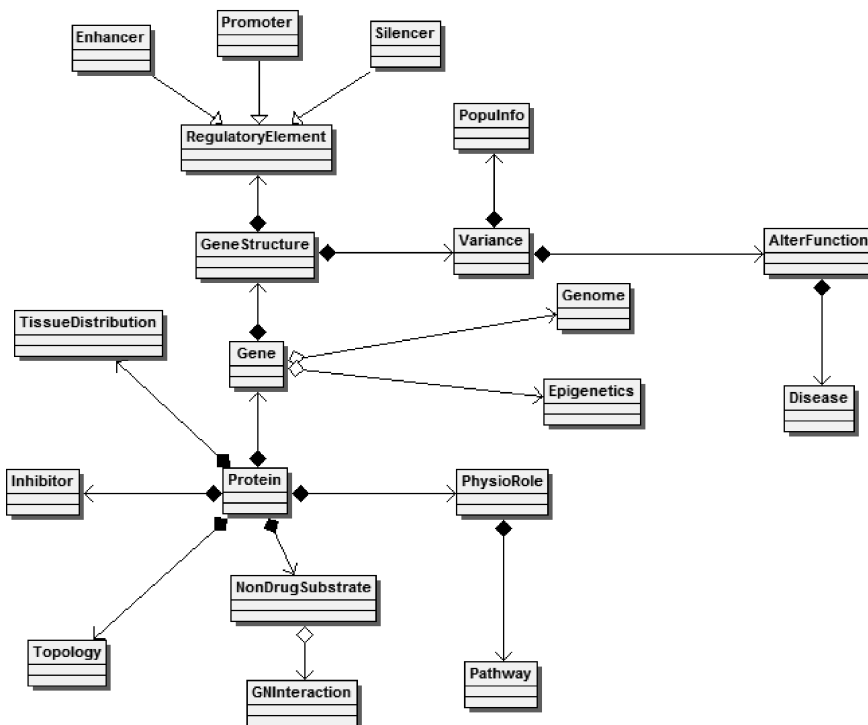


Fig. 2.5. The class diagram of the structure–function correlation.

own attributes. The representation of an association class is useful when there is a relationship between many objects and many other objects, while the attributes are the characteristics of the connection itself but not any of those classes being linked. Such an association class can be connected to the regular association line with a dotted line (*see* Fig. 2.6). A dotted line stands for a dependency. A dependency implies that one of the elements will change when the other changes.

A class hierarchy can be formed when the subclasses inherit attributes and associations of a super class. The subclasses can have particular characteristics of their own. This inheritance relationship between super classes and subclasses is illustrated with an open arrowhead (*see* Fig. 2.5).

Another type of association is aggregation, which refers to the part-whole relationship. An aggregation relationship is represented by a small diamond at the end of the association line that runs between part classes and the whole classes, heading toward the whole class (*see* Fig. 2.5). In an aggregation relationship, the part/child class instance can outlive its parent (the whole class). Such an aggregation relationship can be represented

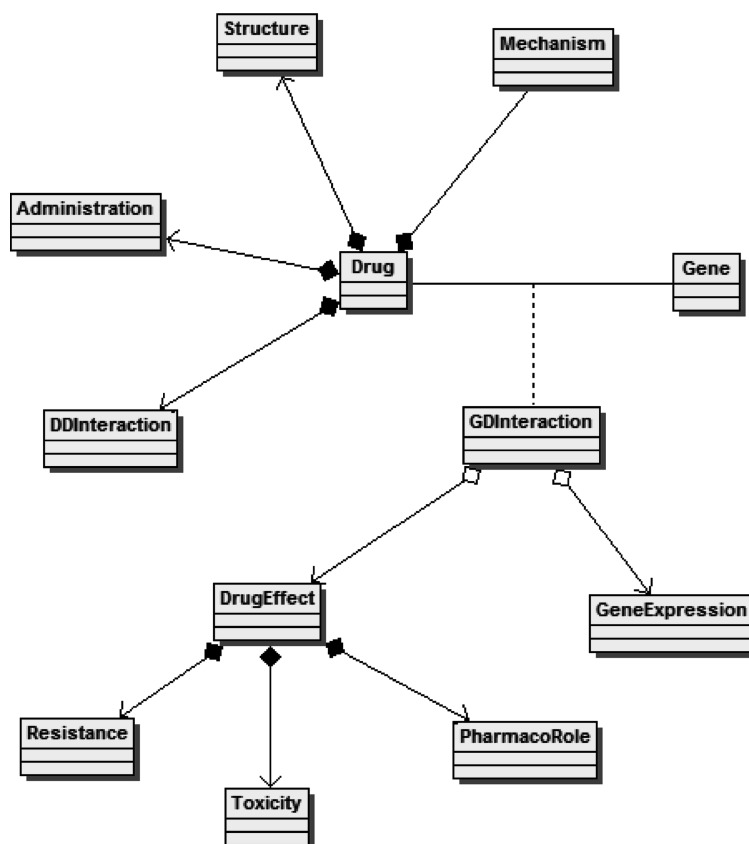


Fig. 2.6. The class diagram of gene-drug interactions.

with an unfilled diamond shape on the parent class's association end. The composition aggregation relationship means the child class's instance is dependent on the parent class's instance lifecycle, which is represented with the filled diamond shape.

3.2.3.1. Data Modeling of the Structure–Function Correlation

As shown in **Fig. 2.5**, gene, protein, and related genomic information are needed to study how the structure affects function. As discussed in **Chapter 1**, the structure–function association is essential for studying all of the emerging knowledge domains, from pharmacogenomics to systems biology, from nutrigenomics to epigenetics.

In this diagram, the classes “Variance,” “Genome,” “Non-DrugSubstrate,” and their associated child classes such as “PopuInfo” and “GNInteraction” represent the features in pharmacogenomics and nutrigenomics. The classes “Disease,” “PhysioRole,” “Pathway,” and “TissueDistribution” are critical for the study of systems biology. The class “Epigenetics” is designed specifically for the emerging field epigenetics and the studies of gene–environment interactions.

The characteristics of “Gene” are described from several aspects and levels including “GeneStructure,” “Protein,” and “Genome.” As illustrated in the biomedical model of **Fig. 2.2**, detailed information of exons, introns, and characteristics of 5' UTR and 3' UTR in the sequence may be included in the “GeneStructure” class.

The function correlated with nucleotide structures (especially regulatory elements) is described in the class “RegulatoryElement.” For example, promoter regions in a gene may influence the gene expression level. The subclasses of regulatory elements include “Enhancer,” “Promoter,” and “Silencer.”

Genetic sequence variation may be crucial in functional variation. Variation is one of the most important features of pharmacogenomics and nutrigenomics, which studies different drug and nutrient responses in individuals. In **Fig. 2.5**, the class “Variance” is used to represent both sequence polymorphisms and mutations, because the definition of the concept “polymorphism” is somewhat narrow (36). The population information (“PopuInfo”) of variations (especially polymorphisms) is used for analysis and selection of certain groups of patients. Sequence variations can alter the transporter function. The altered function and the pathological role of the variation are represented in the class “AlterFunction,” which in turn can result in diseases (“Disease”).

The position of the gene in the whole genome is identified in the “Genome” class. The classes in the protein structure domain include “Protein” and “Topology.” “Topology” includes the domain type in the protein (such as transmembrane domains

(TMDs)). The abundance of transporter proteins in different tissues is described in the class “TissueDistribution.”

The functions correlated with protein structures are described in the classes “PhysioRole,” “NonDrugSubstrate,” and “Inhibitor.” “PhysioRole” describes the physiological functions of the transporter gene, such as their roles in the regulation of intracellular redox potential. The child class “Pathway” put the gene in the whole picture of its functioning processes and interactions. “NonDrugSubstrate” describes non-drug substrates that are known to interact with the transporter, including gene–nutrient interactions (“GNInteraction”). These classes are important for the study of nutrigenomics of transporters (*see Chapter 1*). The details of drug substrates will be described in the gene–drug correlation domain (*see Section 3.2.3.2*). The class “Inhibitor” describes those molecules that can inhibit the transporter function, such as those examples in **Fig. 2.2**.

3.2.3.2. Data Modeling of the Gene–Drug Interaction

The major concepts involved in the gene–drug interaction include gene, drug, and their correlation. Drug information such as drug structure and mechanisms is crucial for the understanding of drug-resistance and toxicity mechanisms. The information can be very helpful for predicting the response of new drugs with similar structure or action. Drug information is also important for designing strategies to reverse the resistance or toxicity associated with side effects and treatment failure. The model representing drug information is illustrated in **Fig. 2.6**.

The classes “Structure,” “Mechanism,” “Administration,” and “DDInteraction” describe several aspects of the major class “Drug.” “Mechanism” contains drug activities and target information. “DDInteraction” means drug–drug interaction. “Administration” describes detailed information of administration of the drug.

The interactions between gene/human and drug in **Figs. 2.2** and **2.3** incorporate two levels of response to drugs, both genotypic and phenotypic. The later level will be discussed in the next section. The class “GDInteraction” describes the overall characteristics and mechanisms of the gene–drug interaction.

Because the gene–drug interaction is mutual, both of the gene and drug can have responsive reactions influenced by each other. Considering the drug side, the actions of a drug may be changed by genetic alterations. For example, increased drug efflux or decreased drug influx may be caused by transporter variations, as illustrated in the biomedical model **Fig. 2.2**. The class “Drug-Effect” in **Fig. 2.6** represents this kind of interaction result. The other side of the gene–drug interaction is “gene,” which is

represented in the class “GeneExpression.” For example, over-expressed breast cancer-resistance protein (BCRP) was found to mediate resistance to mitoxantrone in breast cancer therapy (35).

Figures. 2.5 and **2.6** can be integrated to be one complete model. This can be done with extending and connecting the class “Gene” (in **Fig. 2.6**) to **Fig. 2.5**.

3.2.3.3. Data Modeling of the Genotype–Phenotype Correlation

The genotype–phenotype correlation includes normal, disease, and the drug-response phenotypes. Because the genotype–phenotype correlation is linked to structure–function and gene–drug correlations, the disease and drug-response aspects at this level of correlation are integrated in the two later correlations, as shown in **Figs. 2.5** and **2.6**, respectively. The disease aspect of phenotype is represented in class “Disease” in **Fig. 2.5**. This phenotypic response is correlated with genotypic factors through the correlation with the class “AlterFunction.” The overall drug-response phenotypes are correlated with genotypic gene–drug interactions through class “DrugEffect.” These phenotypes include “Resistance” and “Toxicity,” as illustrated in **Fig. 2.6**. If we abstract the concepts in **Fig. 2.3**, drug activation, inactivation (such as clearance), absorption, distribution (include transportation) can be analyzed and described in the class “PharmacRole,” which describes the pharmacological role in the phenotypic response.

To evaluate this data model and see if it captures the most important aspects in the targeting knowledge domain, it can be checked back with the original biological facts in **Figs. 2.2** and **2.3**. This object model is consistent with the biological model and domain knowledge. For example, most of the objects in **Fig. 2.2**, such as the types of the molecules involved, are represented and included in **Figs. 2.5** and **2.6**. In the implementation of the system, if some modifications are found necessary, the model constructed here can still be changed and improved.

3.2.4. Sequence Diagrams

A sequence diagram uses dynamic views to describe a specific scenario or a real example of a use case (33). Sequence diagrams portray a more detailed view of the interaction between the objects of the main classes in the system. It shows how the actors interact directly with the system, and the system events the actors generate (*see Fig. 2.7*).

In a sequence diagram, the objects identified are listed along the top of the diagram, with a dotted line beneath each object (as shown in **Fig. 2.7**). These dotted lines are called lifelines. The objects are listed in the order they are employed in the scenario. The leftmost object is the one that makes the stimulus that starts the scenario. Events within a sequence, which happen later in the time order, are often represented lower on the chart. Objects are

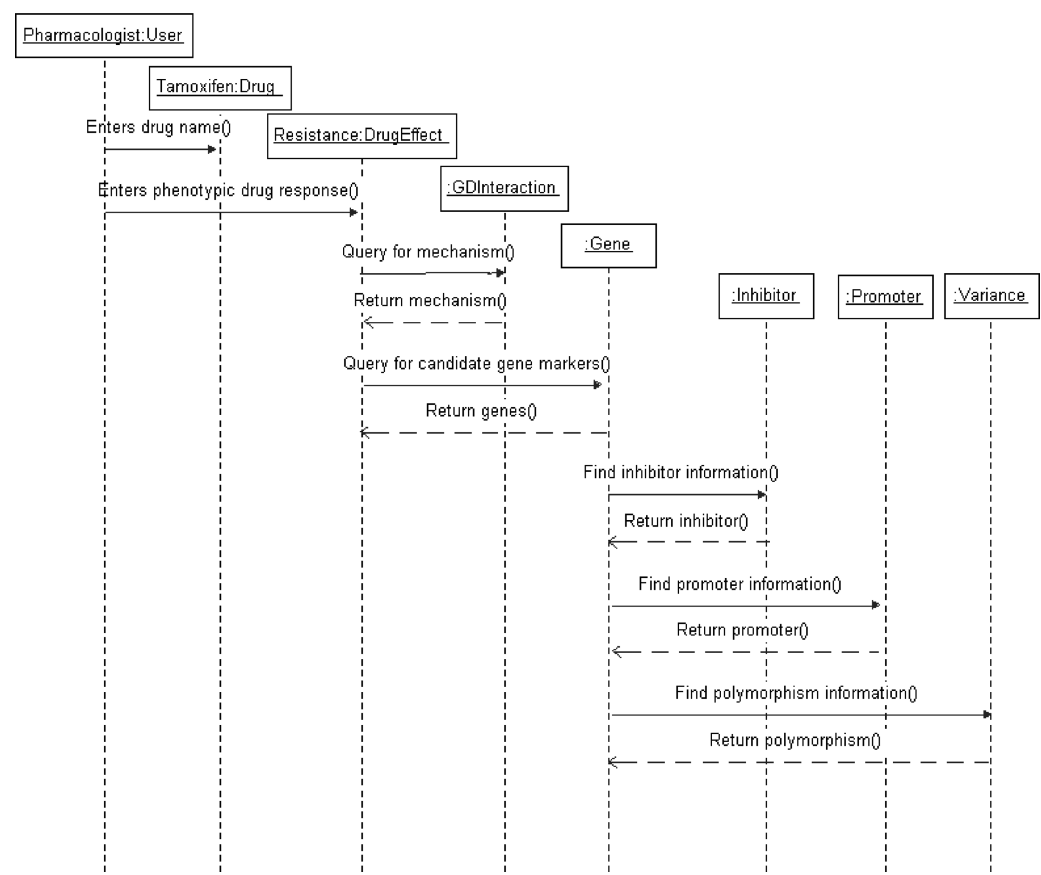


Fig. 2.7. The sequence diagram of the scenario “identifying candidate genetic markers in tamoxifen resistance.”

linked by an event arrow, which suggests that a message is transferring between those two objects. The event ceases at an arrow-head.

Figure. 2.7 shows the sequence diagram of “identifying candidate genetic markers in tamoxifen resistance.” In this scenario, a user, such as a pharmacologist or a drug developer, wants to find the mechanisms and design strategies (such as new drugs) to reverse the resistance to tamoxifen in breast cancer therapy. To do this, they need to know the candidate genes that may be responsible for tamoxifen resistance in breast cancer therapy. These genes can be potential targets for the reversal strategies.

This is a typical scenario with most of the important correlations of pharmacogenomics and systems biology involved (*see Chapter 1*). The goal in this decision-making process is to find *genotypes* that may be responsible for the resistance *phenotype*. With the drug name tamoxifen at the left side as input, what needs to be found out is the other side of the *drug-gene* interaction, the genes. Once the candidate genes and the interaction mechanisms are known, the information about the *structure-function* correlation is needed to identify the possible targets for finding reversal

mechanisms. Such information may include the known inhibitors of the genes, the regulatory elements that affect the transcription activities, as well as variations including genetic polymorphisms for individualized strategies.

As shown in **Fig. 2.7**, the user first queries the system through entering the drug name “Tamoxifen,” whose information is in the “Drug” class. The user then enters the possible phenotypic response “Resistance” to see the possible gene markers. The system looks for the information from the classes “DrugEffect,” “GDIInteraction,” and “Gene,” and extracts the genes that show altered expression as possible markers in the tamoxifen-resistance effect. The user may also want to ask how the genetic factors affect the drug actions, which can be retrieved from the class “GDIInteraction.” Now the user is ready to look for possible reversal targets such as some regulatory elements in the genes that can be used to inhibit the resistance effect. To do this, the information of “Inhibitor” and sequences of “Promoter” can be retrieved. To make the strategies specific for different individuals, the user wants to know the genetic variance that may occur in these genes, which can be obtained from the “Variance” class. With such information, the user can be ready to design reversal strategies.

This scenario describes how the information is extracted for the major correlations, and the information flow among the objects. This example shows how a complicated problem can be solved step by step with a decision support system (34).

4. Conclusion

The object-oriented UML approach provides data modeling capabilities and supports a systematic methodology for pharmacogenomics and systems biology studies in transporters. This methodology helps present data for multiple users including drug designers, pharmacologists, molecular biologists, clinicians, and microarray examination designers. A good methodology not only benefits software developers but also can improve and broaden the applications of the system by the users. It can give analysts the information necessary to make sound decisions about strategic issues for research, drug development, and treatment.

The methodology presented here is an attempt to provide the foundation for a comprehensive system that brings pharmacogenomics and systems biology into the clinic to benefit patients more directly (34). It is not limited for transporter studies but also can be used for other biomedical fields. The model here takes into account of the “variation” beyond the gene sequence, since “sequence variation is only one parameter, and certainly not the dominant parameter in human variation.” (37) By OO modeling using UML, the problems of transporter pharmacogenomics and

systems biology can be approached from different angles with a more complete view, which may greatly enhance the efforts in effective drug discovery and development.

The construction of the model demonstrates that UML, a modeling language that has been widely used in the business information technology (IT) industry, can be an appropriate common literacy in biomedicine if applied with appropriate methodologies. The main difficulty of applying UML in the biomedical domain is the barrier of domain knowledge. The “requirement analysis and biomedical models \Rightarrow concept abstraction and object design \Rightarrow UML OO models” methodology developed here is a useful measure for knowledge modeling in biomedicine. This methodology provides a generic way in how to build a computer OO model from the crude domain knowledge.

In biomedical science, knowledge modeling could play the role of mathematical modeling in physical sciences (18). Just as mathematicians, biomedical informaticians study models that are abstractions from the real biomedical world. As mathematical modeling encodes knowledge in a dynamic physical system, knowledge modeling (as the examples shown here) in biomedical systems also embraces both structural and dynamic behavioral aspects. This characteristic allows the dynamic representation of interactions and can be especially useful for the study of structure–function, gene–drug, and genotype–phenotype associations in transporter pharmacogenomics and systems biology (*see Chapter 1*).

References

1. Pipas, J.M. and McMahon, J.E. (1975) Method for predicting RNA secondary structure. *Proc. Natl. Acad. Sci. USA* **72**, 2017–2021.
2. Studnicka, G.M., Rahn, G.M., Cummings, I.W., Salser, W.A. (1978) Computer method for predicting the secondary structure of single-stranded RNA. *Nucleic Acids Res.* **5**, 3365–3387.
3. Erdmann, V.A. (1978) Collection of published 5S and 5.8S ribosomal RNA sequences. *Nucleic Acids Res.* **5**, r1–r13.
4. Dayhoff, M.O., Schwartz, R.M., Chen, H.R., Hunt, L.T., Barker, W.C., Orcutt, B.C. (1980) Nucleic acid sequence bank. *Science* **209**, 1182.
5. Korn, L.J., Queen, C.L., Wegman, M.N. (1977) Computer analysis of nucleic acid regulatory sequences. *Proc. Natl. Acad. Sci. USA* **74**, 4401–4405.
6. McCallum, D. and Smith, M. (1977) Computer processing of DNA sequence data. *J. Mol. Biol.* **116**, 29–30.
7. Fuchs, C., Rosenvold, E.C., Honigman, A., Szybalski, W. (1978) A simple method for identifying the palindromic sequences recognized by restriction endonucleases: the nucleotide sequence of the AvaII site. *Gene* **4**, 1–23.
8. Gingeras, T.R., Milazzo, J.P., Roberts, R.J. (1978) A computer assisted method for the determination of restriction enzyme recognition sites. *Nucleic Acids Res.* **5**, 4105–4127.
9. Paulsen IT, Nguyen L, Sliwinski MK, Rabus R, Saier MH Jr. (2000) Microbial genome analyses: comparative transport capabilities in eighteen prokaryotes. *J. Mol. Biol.* **4**, 75–100.
10. Chen, L., Ortiz-Lopez A., Jung A., Bush D.R. (2001) ANT1, an aromatic and neutral amino acid transporter in Arabidopsis. *Plant Physiol.* **125**, 1813–1820.
11. Dawson, P.A., Mychaleckyj, J.C., Fossey, S.C., Mihic, S.J., Craddock, A.L., Bowden, D.W. (2001) Sequence and functional

- analysis of GLUT10: a glucose transporter in the Type 2 diabetes-linked region of chromosome 20q12-13.1. *Mol. Genet. Metab.* **74**, 186–199.
12. Kihara, D. and Kanehisa, M. (2000) Tandem clusters of membrane proteins in complete genome sequences. *Genome Res.* **10**, 731–743.
 13. Yan, Q. (2001) *Informatics Support for Human Membrane Transporter Pharmacogenomics Studies*. ProQuest, Ann Arbor, MI, pp. 1–138.
 14. <http://tcdb.ucsd.edu/index.php> (accessed in May 2009).
 15. Nebert, D.W. (1999) Pharmacogenetics and pharmacogenomics: why is this relevant to the clinical geneticist? *Clin. Genet.* **56**, 247–258.
 16. Sissung, T. M., Gardner, E. R., et al. (2008) Pharmacogenetics of membrane transporters: a review of current approaches. *Methods Mol. Biol.* **448**, 41–62.
 17. Frishman, D., Heumann, K., Lesk, A., Mewes, H.W. (1998) Comprehensive, comprehensible, distributed and intelligent databases: current status. *Bioinformatics* **14**, 551–561.
 18. Rechenmann, F. (2000) From data to knowledge. *Bioinformatics* **16**, 411.
 19. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Rumbaugh, J., Lorenson, W. (1991) *Object-Oriented Modeling and Design*. Prentice Hall, pp. 1–500.
 20. Korson, T. and McGregor, J. (1990) Understanding Object-Oriented: A Unifying Paradigm. *CACM* **9**, 40–60.
 21. Object Management Group. (1999) *OMG Unified Modeling Language Specification*. Object Management Group, Inc., pp. 1–808.
 22. Martinez, R., Rozenblit, J., Cook, J.F., Chacko, A.K., and Timboe, H.L. (1999) Virtual management of radiology examinations in the virtual radiology environment using common object request broker architecture services. *J. Digit. Imaging* **12**, 181–185.
 23. Quitadamo, L. R., Marciani, M. G., et al. (2008) Describing different brain computer interface systems through a unique model: a UML implementation. *Neuroinformatics* **6**, 81–96.
 24. Fridsma, D. B., Evans, J., et al. (2008) The BRIDG project: a technical report. *J. Am. Med. Inform. Assoc.* **15**, 130–137.
 25. Webb, K. and White, T. (2005) UML as a cell and biochemistry modeling language. *Biosystems* **80**, 283–302.
 26. Taylor, C. F., Paton, N. W., et al. (2003) A systematic approach to modeling, capturing, and disseminating proteomics experimental data. *Nat. Biotechnol.* **21**, 247–254.
 27. Xirasagar, S., Gustafson, S., et al. (2004) CEBS object model for systems biology data, SysBio-OM. *Bioinformatics* **20**, 2004–2015.
 28. Jedlitschky, G., Leier, I., Buchholz, U., Barnouin, K., Kurz, G., and Keppler, D. (1996) Transport of glutathione, glucuronate, and sulfate conjugates by the MRP gene-encoded conjugate export pump. *Cancer Res.* **56**, 988–994.
 29. Hipfner, D.R., Deeley, R.G., and Cole, S.P. (1999) Structural, mechanistic and clinical aspects of MRP1. *Biochim. Biophys. Acta.* **1461**, 359–376.
 30. Hyde, S.C., Emsley, P., Hartshorn, M.J., et al. (1990) Structural model of ATP-binding proteins associated with cystic fibrosis, multidrug resistance and bacterial transport. *Nature* **346**, 362–365.
 31. Cui, Y., Konig, J., Buchholz, J.K., et al. (1999) Drug resistance and ATP-dependent conjugate transport mediated by the apical multidrug resistance protein, MRP2, permanently expressed in human and canine cells. *Mol. Pharmacol.* **55**, 929–937.
 32. Keppler, D., Leier, I., Jedlitschky, G., and Konig, J. (1998) ATP-dependent transport of glutathione S-conjugates by the multidrug resistance protein MRP1 and its apical isoform MRP2. *Chem. Biol. Interact.* **111–112**, 153–161.
 33. Harmon, P. and Watson, M. (1997) *Understanding Uml: The Developer's Guide: With a Web-Based Application in Java*. Morgan Kaufmann Publishers, pp. 1–340.
 34. <http://pharmtao.com/transporter> (accessed in May 2009).
 35. Ross, D.D., Yang, W., Abruzzo, L.V., et al. (1999) Atypical multidrug resistance: breast cancer resistance protein messenger RNA expression in mitoxantrone-selected cell lines. *J. Natl. Cancer Inst.* **91**, 429–433.
 36. Yan, Q., and Sadée, W. (2000) Human membrane transporter database: a Web-accessible relational database for drug transport studies and pharmacogenomics. *AAPS PharmSci* **2**, E20.
 37. Marshall, A. (1997) Laying the foundations for personalized medicines. *Nat. Biotechnol.* **15**, 954–957.



<http://www.springer.com/978-1-60761-699-3>

Membrane Transporters in Drug Discovery and
Development

Methods and Protocols

Yan, Q. (Ed.)

2010, XIII, 365 p., Hardcover

ISBN: 978-1-60761-699-3

A product of Humana Press