A BIOMEDICAL KNOWLEDGE DISCOVERY IN DATABASES DESIGN TOOL - TURNING DATA INTO INFORMATION

Pfeifer B[1], Tejada MM[1], Kugler K[2], Osl M[1], Netzer M[1], Seger M[1], Modre-Osprian R[3], Schreier G[3], Tilg B[1]

Abstract

We present a toolbox for bridging the back room with the front room of a data warehouse. The user can generate statistical or data mining workflows in order to better understand diseases or to discover new biomarker candidates. Knowledge discovery is an iterative process, where different processing steps have to be assembled to functional objects by a specialist. Designing appropriate workflows enables to solve a defined question. These workflows can easily be used for ulterior purposes by only adding new data and by parametrizing the functional objects. To tackle this problem the toolbox is split up into two parts. The first part consists of a graphical designer tool. The second part is a runner application that enables the operator to execute previously defined workflows. Intelligent query models in Marfan's syndrome were implemented. Thereby, the identification of genotype-phenotype correlations was enabled.

1. Introduction

A data warehouse is a central collection or repository for persistently storing all analysis relevant data and information [1, 2, 3]. Scattered and smothered under gigabytes and terabytes of data, useful information may be hidden. Data warehouses coupled with intelligent search, data mining approaches and data discovery toolboxes enable to collect and process these data in order to turn them into useful information and new knowledge [2, 4, 5]. As a matter of fact, it is impossible to buy a data warehouse like a standard application. The conceptual design and the implementation are dependent, among other things, on the enterprise size and the used systems. Therefore, the development of a data warehouse can shape up at a tedious and costly process. The correct answering of questions like which data the employee or scientist need, which impact does the data have, in which aggregated form and formats the data have to be delivered, which sources are available and how to access the data is of importance for completing a data warehouse project. In contrast to transactional systems a data warehouse represents a global view on the data. Furthermore, the amount of data stored in a data warehouse can be tremendously big due to the long-term period storage of data in order to perform business intelligence tasks.

[1] UMIT / Institute of Biomedical Modeling / 6060 Hall, Austria
[2] UMIT / Institute for Bioinformatics / 6060 Hall, Austria
[3] eHealth systems, Austrian Research Centers GmbH – ARC, 8020 Graz, Austria

Modern techniques, such as electronic data processing, enable the usage of information as a resource. Consequently, classical goods are rapidly being thrust aside by the intelligence, which was settled to goods in order to create it. The information age started in the mid seventies and became essential in our modern way of living. This statement forces us to define what information is and how information comes into being.

Data can be described as logical grouped information units and are therefore the fundamental components of information [6]. In the field of computer sciences data are understood as machine readable and processible digital representations of information. The information itself is character coded, and the information is generated using defined rules. Therefore, information follows syntax. In order to extract information from data, those have to be interpreted in a semantic context.

Thus, the data warehouse itself can be said to be the basis for all further steps. This back room component, where only the database and the data warehouse administrator has direct access, needs to be bridged with the front room. Users and scientists have access to the data via the front room for generating new knowledge and systems biology approaches without having to know database query languages like e.g. SQL. An integrated business intelligence application or toolbox is essential because it cannot be assumed that an average user is familiar with the interaction of statistical, data mining or generally knowledge discovery approaches. Furthermore, one common mistake is declaring a spreadsheet application, which can be used for basic tasks like visualization, to be the main knowledge discovery tool. To overcome the above-mentioned problems a Knowledge Discovery in Databases Designer (KD^3) has been implemented.

2. Methods & Results

A systematic approach for data mining starts with elaborating the conceptual formulation followed by data analysis and modeling, validation of the process and delivery of the results. Therefore, one can say, that many different steps are involved in a classical knowledge discovery approach [7]. The main steps are focusing the problem, preprocessing and transformation of the data, performing the data mining or statistical approach, and evaluation. When a data warehouse is used as the basis, data are delivered in an integrated and consistent form, which reduces the preprocessing and transformation process dramatically. Hence, a data-mining specialist can focus on assembling different approaches in order to generate new knowledge or answering a specified question. Due to the reason that usually more than one approach is needed in order to solve a specified problem, those have to be coupled together by designing a workflow.

2. 1. KD^3 Composition

The implemented KD^3 application consists of four main parts. A screenshot of the application is depicted in *Figure 1*. The application divides the screen into four parts. The first part shows the available functional objects or tasks, which are loaded using the Java reflection API and are grouped using a hierarchical structure. In the workspace window (2) the user can drop functional objects and parameterize them by setting up the constructor. The constructor is the main function that is called by the system when instantiating the object. Therefore, the constructor is used for initializing the object with default parameters or for querying the objects result set. In the window section (3) the workflow is visualized using graph-ml, which is also used for storing and deploying workflows. The window section (4) is used for displaying messages, which occur during processing a workflow. Furthermore, the user can display temporary or transient results.

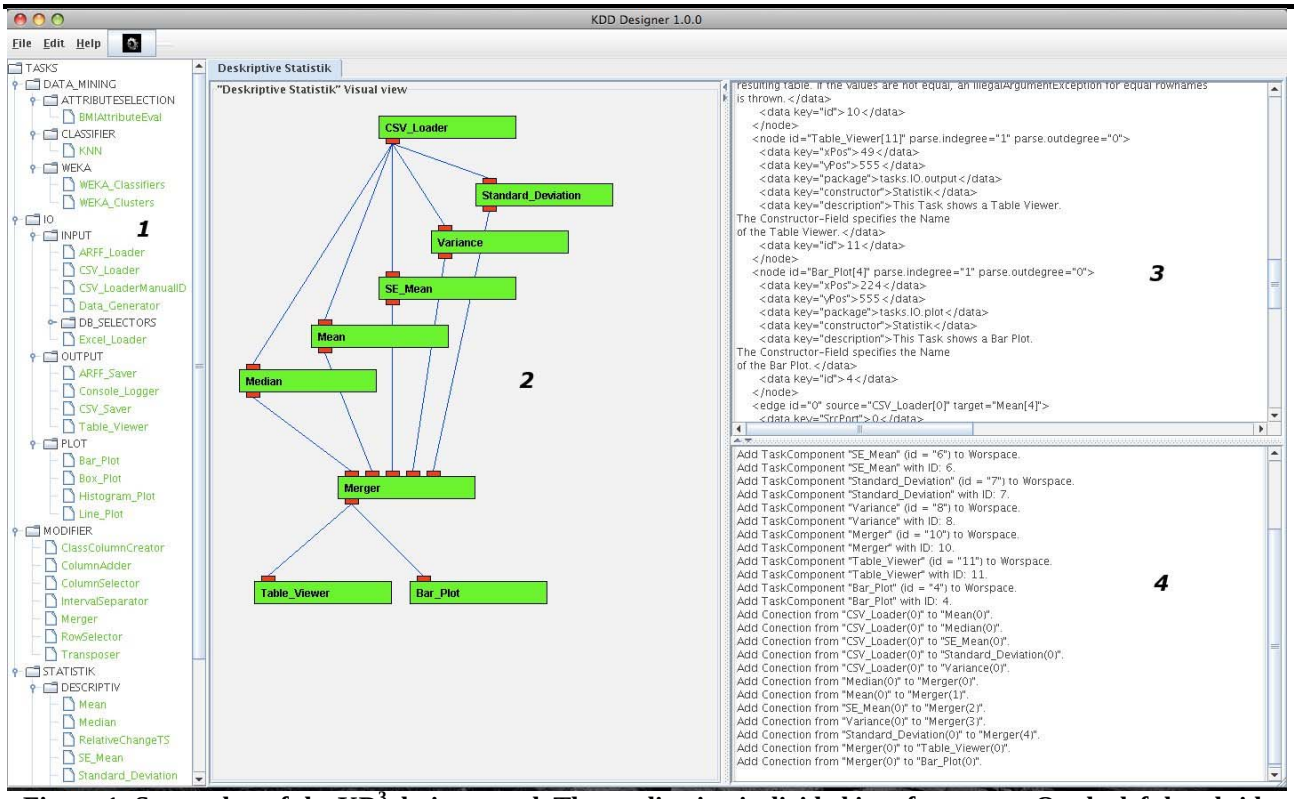

Figure 1: Screenshot of the KD^3 designer tool. The application is divided into four parts. On the left hand side (1) all functional objects are grouped into packages, in the section (2) the designed workflow is visualized and can be parameterized by the user, screen section (3) displays the workflow using GraphML and the windows section (4) is for logging and testing purpose.

2. 2. KD^3 Tasks

A task or functional object is composed as following: From the users perspective a task is a functional object, which consists of in- and out-ports that have to be assembled and parameterized to fulfill their purpose (e.g. computing, querying a database, interacting with other objects). From the software engineer's point a task object is derived from the super class named `KDDTask`. A KD^3 Task consists of a description field, where information how to configure the constructor has to be set. Furthermore, the above described in- and out-ports, have the job to get the data from another object and to send the processed data to another object. For performing the computation an abstract method named `compute()` is available. This method must be overridden in derived task classes in order to solve a specified problem.

2. 3. KD^3 Workflow

A workflow is characterized as a repeatable pattern activity, which is used to solve a defined process using different parameters and data sets. Therefore, the data miner has to drop the functional objects into the workspace window (windows section (2) depicted in *Figure 1*). After that the objects constructor needs to be set by double clicking on the functional object. *Figure 2* depicts one example (a functional merger object) of a parametrizing window.

After the workflow is designed it can be executed using the execute function of the KD^3 designer. When the workflow is properly designed it can be saved as GraphML [14] file and deployed e.g. to biomedical researchers in order to analyze newly available data.

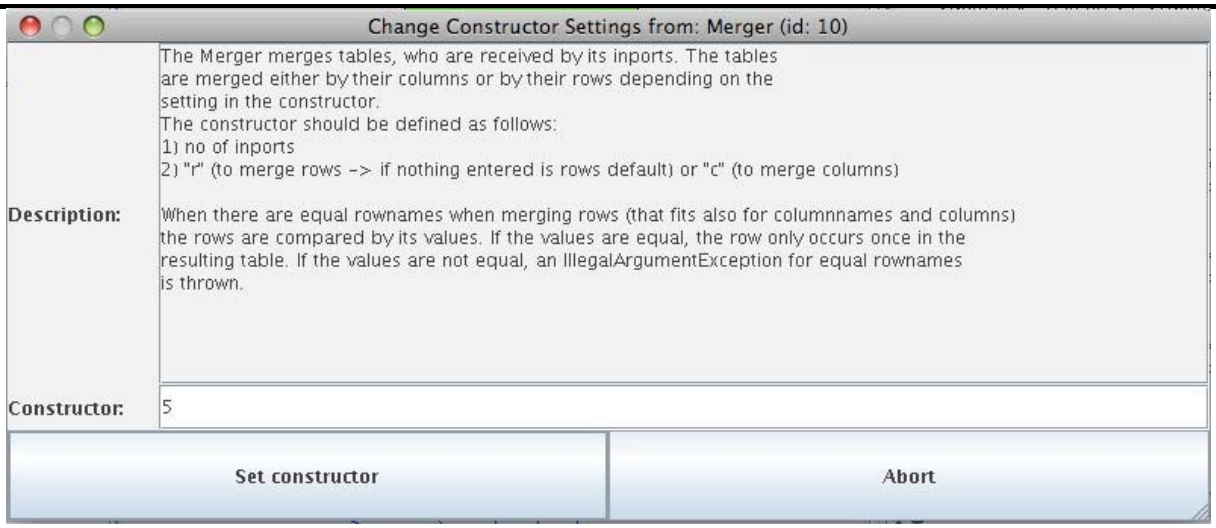

Figure 2: Window for parametrizing a functional object using its constructor. The constructor argument is set as a string object, which is then interpreted by the KD^3 designer.

2. 4. Sample workflow using KD^3: intelligent query models for genotype-phenotype correlation in patients with FBN1 mutations

Mutations in the fibrillin-1 coding gene FBN1 have been shown to cause Marfan's Syndrome (MFS), a multisystemic connective tissue disorder with autosomal dominant trait [8]. This abnormal genetic condition is characterized by pleiotropic manifestations involving predominantly the ocular, skeletal, and cardiovascular systems [9]. Many studies have been focused on exploring the correlations between mutations in the FBN1 gene and the clinical phenotype, as they are crucial in predicting the clinical consequences of a specific mutation. In the absence of solid correlations the identification of an FBN1 mutation has little prognostic value [10]. Nevertheless, this can be a very challenging task due to the pleiotropic and age related nature of the disease [11].

Therefore, functional bioinformatics objects and tasks for the KD^3 designer have been developed to systematically study and support genotype-phenotype correlations in patients with Marfan's Syndrome by intelligent query processing. For accessing the underlying data warehouse system four different levels of mutational information were defined. The data warehouse contains 436 entries with 372 different mutations. 361 constitute substitutions and 75 non-substitutions. Of the substitutions, 272 are missense mutations, 48 nonsense mutations, and 43 splice site mutations. The group of non-substitutions can be split up in 54 deletions and 21 insertions.

In the implemented model disparate mutation classes were defined at four different levels of mutational information: 1) Mutation type, 2) Mutation type and consequence, 3) Mutation type, consequence and location, and 4) Mutation type, consequence and amino acid changes. For this task several factors were taken into account, for instance, the age at onset and the involvement of major and minor criteria, using the known genotype-phenotype correlations as reference points.

The developed models enable the assignment of a clinical phenotype of unknown genotype to a mutation class at each level of information according to the type of mutation, the consequence at the protein level, the location of the mutational event, and the amino acid changes. Score values based on the relative entropy of the probability distributions of two mutation classes were calculated for each clinical symptom. Four decision rules built on square distance measures and several weighting factors deduced from a medical knowledge base were defined to assign a query pheno-

type to the corresponding mutation class. The results demonstrate that the proposed approach is well suited for studying and identifying genotype-phenotype correlations in MFS.

To process a query phenotype three preprocessing steps have to be executed: In the first step a frequency matrix has to be generated, followed by the calculation of score values for each considered clinical feature represented as a matrix of score sets, which have to be balanced in the last preprocessing step. After accomplishing these steps the actual query execution can begin, enabling the assignment of a query phenotype of unknown genotype to a mutation class in the database at each information level. *Figure 3* depicts the workflow assembled with the various tasks from the framework.

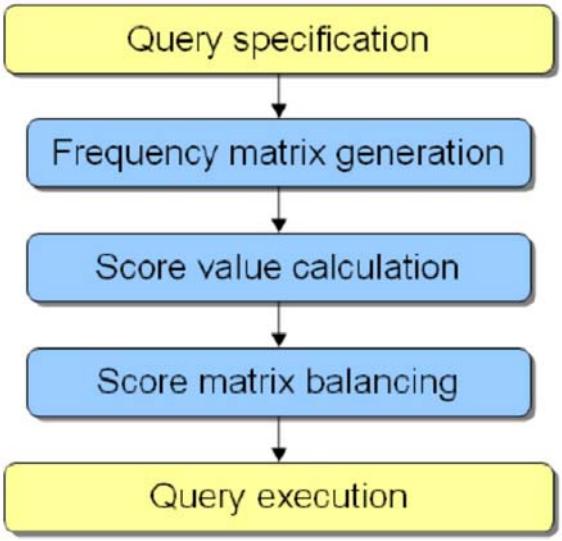

Figure 3: Flow of the intelligent query processing

Using the proposed and implemented workflow enabled the identification of genotype-phenotype correlations, which is of crucial importance to the prediction of clinical consequences of FBN1 mutations.

Every patient with Marfan's Syndrome is at risk of developing severe complications at any of the four affected organ systems. The diagnosis, based solely on clinical criteria, can be complemented by molecular genetic testing. This could make a potential contribution towards establishing the family history as an unequivocal major criterion for diagnosis or confirming an apparent de novo mutation, which is important for the counseling of at-risk relatives [12]. Furthermore, it could be very useful for an early diagnosis of the disease. However, without any solid genotype-phenotype correlations the identification of a mutation has little prognostic value. The proposed bioinformatics KD^3 tasks enable a non-database and data mining expert to systematically study and support genotype-phenotype correlations in patients with MFS.

3. Discussion & Conclusion

The KD^3 designer enables the data miner or business intelligence specialist to build defined workflows in order to turn data into information to gain new knowledge for a better understanding of disease mechanisms. Apart from this, the finding of unknown patterns and biomarker candidates or helping to enhance medical treatment is also targeted. The developed and tested workflow can then be deployed and used by other researchers and medical staff.

Moreover, it is possible to integrate external frameworks and toolboxes by implementing the defined adaptor specified in the super class of the framework. As one example we integrated the well-known data-mining framework WEKA [13] into the KD^3 designer.

Motivated by biomedical research projects an easy to use and easy expandable system was developed, which is used in our biomedical research projects. The KD^3 designer helps to overcome the problem, that business intelligence tools exists in a various number but are not well integrated into data warehouse projects. The toolbox bridges the gap between the back room and the front room component of a biomedical data warehouse project.

4. Acknowledgement

The Federal Ministry of Economics and Labour of the Republic of Austria, the Tyrolean Future Foundation and the Competence centre HITT-health Information Technologies Tyrol funded this work.

5. References

[1] R KIMBALL, J CASERTA. The Data Warehouse ETL Toolkit, Wiley Publishing (2000)

[2] A. BAUER, H. GUNZEL. Data Warehouse Systeme. Dpunkt Verlag (2004)

[3] Bloor: Data Warehousing Tools and Solutions. IT-Verlag (1997)

[4] PFEIFER B, ASCHABER J, BAUMGARTNER CH, DREISEITL S, MODRE-OSPRIAN R, SCHREIER G, TILG B. A data warehouse for prostate cancer biomarker discovery, BioComp 2007. Las Vegas, USA, 2007. Vol 2: p 316-323

[5] PFEIFER B, ASCHABER J, BAUMGARTNER C, DREISEITL S, MODRE R, SCHREIER G, TILG B. A Life Science Data Warehouse System to enable Systems Biology in Prostate Cancer. 4th International Workshop, p 9ff. DILS 2007, Pennsylvania, USA. 2007.

[6] R KIMBALL, M ROSS. The Data Warehouse Toolkit, Wiley Publishing (2002)

[7] U FAYYAD, G PIATESKY-SHAPIRO, P SMYTH. Knowledge Discovery and Data Mining. Proceeedings of the Second International Conference on Knowledge Discovery. 1996

[8] D.J. HALLIDAY, S. HUTCHINSON, L. LONIE, J.A. HURST, H. FIRTH, P.A. HANDFORD, AND P WORDSWORTH. Twelve novel FBN1mutations in Marfan syndrome and Marfan related phenotypes test the feasibility of FBN1 mutation testing in clinical practice. J. Med. Genet., 39(8):589-593, 2002.

[9] P. COMEGLIO, P. JOHNSON, G. ARNO, G. BRICE, A. EVANS, J. ARAGON-MARTIN, F. PEREIRA DA SILVA, A. KIOTSEKOGLOU, AND ANNE CHILD. The Importance of Mutation Detection in Marfan's Syndrome and Marfan-Related Disorders: Report of 193 FBN1 Mutations. Human Mutation, 28(9):928, September 2007.

[10] C. BAUMGARTNER, G. MATYAS, B. STEINMANN, M. EBERLE, J. STEIN, AND D. BAUMGARTNER. A bioinformatics framework for genotype-phenotype correlation in humans with Marfan syndrome caused by FBN1 gene mutations. J. of Biomedical Informatics, 39(2):171-183, 2006.

[11] D.P. JUDGE AND H.C. DIETZ. Marfan's Syndrome. The Lancet, 366:1965, 1976, 2005.

[12] A. DE PAEPE, R.B. DEVEREUX, H.C. DIETZ, R.C. HENNEKAM, AND R.E. PYERITZ. Revised diagnostic criteria for the Marfan syndrome. Am. J. Med. Genet., 62:417-426, 1996.

[13] WEKA, http://www.cs.waikato.ac.nz/~ml/weka/index.html

[14] GraphML, http://graphml.graphdrawing.org/