ILP Challenge 2005: The Safarii MRDM environment

Arno J. Knobbe[1,2], Eric K. Y. Ho[1], Rainer Malik[2]

[1]Kiminkii, Postbus 171, NL-3990 DD, Houten, The Netherlands
a.knobbe@kiminkii.com, e.ho@kiminkii.com
[2]Utrecht University, P.O. box 80 089, NL-3508 TB Utrecht, The Netherlands

Abstract In this paper, we provide some preliminary result for the ILP Challenge 2005 concerning a genetic database of information related to the function of a range of yeast genes. The yeast database consists of multiple tables, and is hence a multi-relational problem. We demonstrate how the MRDM packages Safarii (mining) and ProSafarii (pre-processing) can be used to mine this data. We provide biological justification for the results obtained.

1 Introduction

This paper considers the genetic database of genes that make up the yeast genome *Saccharomyces cerevisiae*, which is provided as a challenge for data miners in the context of the ILP 2005 conference [2]. This database contains descriptions of individual genes, and lots of background information such as the homology between pairs of yeast genes, secondary structure information and homology with different genes that appear in a database known as SwissProt. The database thus describes structured information, which makes the analysis multi-relational. The data is spread over a total of 11 tables.

In this paper we describe a Data Mining exercise based on the Multi-Relational Data Mining framework implemented in the Safarii package, developed by the first two authors. Safarii provides a number of algorithms that work on multi-relational data stored in a relational database. The mining package is supported by a pre-processing tool known as ProSafarii. In the next section, we give an overview of how our MRDM framework and the two software packages work. Section 3 gives an overview of the structure of the yeast database. In Section 4 we provide a number of Data Mining settings that were tried, as well as the required pre-processing involved. Section 5 describes the results for the different settings. We give some biological interpretation of the results as well.

2 Multi-Relational Data Mining & Safarii

Safarii implements a style of MRDM introduced in [3, 4]. It assumes all data is stored in a relational database containing multiple tables. One of the tables (the *target table*) represents the primary entity that is being analysed. Each individual is represented by a single record in the target table. Structured data related to the individual can be looked up by following the associations between tables (a form of

foreign key relations). Class values pertaining to individuals also appear in the target table.

Unlike most ILP systems, Safarii does not use Prolog to describe discovered knowledge (nor the data itself for that matter). Rather, it uses a graphical language known as *selection graphs* [3], which captures both the structural properties of specific individuals, as well as attribute conditions on specific parts of the structure. Selection graphs can be easily translated to first-order logic or SQL, and thus form a bridge between MRDM and relational databases. In essence, selection graphs describe subgroups of the database, and can thus be seen as building blocks for more complicated models of the data, such as decision trees.

Safarii relies to a large extent on the computational power of the RDBMS that stores the data. Rather than loading all the data in main memory, it employs a small set of data mining primitives (predefined templates of SQL queries) for access to the data. Each primitive produces the required statistics for a small set of similar multi-relational hypotheses. Based on these statistics, the actual mining algorithm determines the quality of specific hypotheses, and decides which ones to consider for further refinement. Hence the mining component of the system only deals with relatively compact summaries of the data, and important issues of scalability and query optimisation are left to the RDBMS.

Most relational databases have a well defined data model. Safarii uses this data model as declarative bias. The data model is provided in the form of an XML dialect known as MRML (Multi-Relational Modelling Language). An MRML file contains in textual form the graphical structure of the database, as well as names and types of the different attributes available. Hence the file is the textual equivalent of UML Class Diagrams which can be used to model relational databases [3]. An example of the graphical structure of the ILP 2005 Challenge database can be seen in Figure 1.

Safarii supports two classes of mining algorithm. The first, Rule Discovery, produces a set of independent rules with a fixed target concept, based on one of the target attributes in the target table. Rules can be discovered with a range of rule measures and search strategies. The second algorithm implements a Separate & Conquer strategy for building multi-relational decision lists. The decision list can be used as a classifier (unlike the rule set produced by Rule Discovery). The Separate & Conquer algorithm actually encapsulates the Rule Discovery algorithm. Each decision in the decision list is based on the best rule discovered by Rule Discovery, given the current context of individuals.

The Safarii mining environment comes with a pre-processing companion called ProSafarii. This tool performs a limited form of reasoning about the data model and the database, and subsequently suggests a number of tentative transformations. It contains 9 classes of transformation that are potentially useful in the context of MRDM and Safarii specifically, ranging from denormalisation to aggregation and multi-relational discretisation. After validation by the user, ProSafarii performs the actual transformations and writes a corresponding MRML file. The result can hence be mined directly by Safarii.

3 ILP 2005 Challenge Data

The genetic data that is the subject of the Challenge is provided through [2]. The original data consists of a large collection of Prolog files with facts for some 15 predicates. A fair amount of work was required to parse this data, and load it into a relational database. Before actually mining the data some more pre-processing of the data was required (as is explained in the next section). The data model of the initial database is shown in Figure 1. This database relates to the original Prolog modelling as follows. Predicate names appear in `Courier`, and table names in **bold**.

The primary entity of the relational database is **gene**. This table contains the 6000 odd genes that make up the yeast genome *Saccharomyces cerevisiae*. The gene entity does not appear directly in the data, but gene identifiers can be extracted from `yeast-labelled` and `-unknown`, or `alpha_dist`, `beta_dist` or `coil_dist`. The latter three predicates also have been normalised onto **gene**. The many-to-many relationship between genes and functional classes is implemented by the **hasfunction** table, which connects **gene** to **funcat**, a list of functional categories with descriptions. The internal hierarchical structure of category identifiers (e.g. 01.02.01 *nitrogen and sulfur utilization* falls under 01.02 *nitrogen and sulfur metabolism*, which again falls under 01 *METABOLISM*) has been made more accessible by adding attributes in **funcat** for each of the five levels.

The **homology** table describes the homology between pairs of yeast genes in terms of an *e*-value. Only a fraction of the more than 18 million possible pairs occur. Similar homology data (in the **eval** table) is available between yeast genes and genes occurring in a database called SwissProt. This homology data is also incomplete. Again, identifiers for the SwissProt genes can be extracted from multiple predicates, for example `classification`, `mol_wt` or `sq_len`. This information is gathered in the **swissprot** table.

The **class** table (derived from `parent`) serves as a look-up for the classification of the organism and its position in a taxonomy of species. Different classes of species can be accesses by recursively following the is-a relation within **class** (e.g. *saccharomyces* is-a *saccharomycetaceae* is-a … *fungi*, etc.). Two further tables complement the SwissProt information. The **keyword** table lists keywords for the SwissProt protein. The **databaseref** table lists databases that specific proteins appear in.

The secondary protein structure of yeast genes is provided in the **secondary** table. Each entry describes a secondary structure element for a specific gene, with details such as the type (a, b or c), length and order. Some aggregation on the secondary structure is provided by the attributes *alpha*, *beta* and *coil* in the **gene** table.

Additionally we have employed a software tool for computing secondary structure homology. Ssea [1] is a program which computes alignments of protein secondary structures. It can compute either local or global alignments, with the latter usually aligning longer stretches at the cost of lower overall similarity. Therefore, we include both scores (global and local) in the database.

The following tables summarises the amount of records in each of the tables.

gene	4,053
hasfunction	12,839
funcat	1,307
homology	1,044,816
eval	3,618,919
swissprot	46,163

databaseref	196,535
keyword	14,270
class	3,105
secondary	384,165
ssea	8,211,378

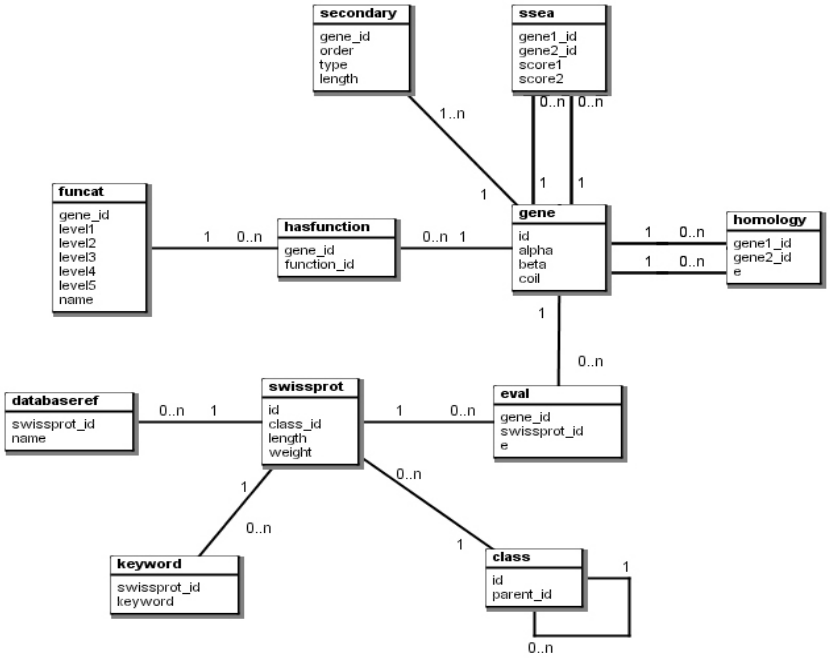


Figure 1 Data model of the ILP 2005 Challenge database.

4 Approach

From a Data Mining standpoint, the most striking feature of the database is the lack of a single clearly defined target attribute. The target concept, the function of a gene, is set-valued: genes may have zero or more functions from a total list of 1307 defined functional classes. To complicate things even more, the functional classes are not independent, but organised in a 5 level hierarchy known as FunCat. Hence, if a gene has a certain specific function that is at a leaf of the hierarchy (no subclasses), one could argue that it also belongs to any functional class that is an ancestor of the specified class.

This functional classification scheme allows for a great many target concepts. Specific functions however may be very rare, and any relation to very similar functions (such as siblings in the hierarchy) will be lost. As an alternative, we have

opted for treating the classes on the highest level (level 1) in the hierarchy as target concepts. If a gene has a function that is a descendent of a specific level 1 class, we treat this gene as a positive example of that class. Of the 28 level 1 classes that appear in the FunCat hierarchy, 19 feature in the `yeast.labelled` sample (including surprisingly the function 99 *unknown*). We therefore will assume 19 target concepts from now on. We would like to stress however that adding any other FunCat related target concept would be just as easy, were one so inclined.

As was mentioned, Safarii requires any target concepts to appear as attributes in the target table (**gene**). A first step therefore is to preprocess the tables **funcat**, **hasfunction** and **gene**. We start by denormalising **funcat** on **hasfunction**, with the aid of ProSafarii. The effect is a replication of class information (specifically *level1*) in the **hasfunction** table. We then apply a so-called reverse pivot transformation to produce the 19 additional binary attributes in **gene** corresponding to each of the 19 level 1 classes.

The resulting **gene** table, primarily containing binary data, leads to a first opportunity for Data Mining. Using Safarii, we can search for association rules within the 19 functional classes. This analysis ignores any information outside the target table, and is hence propositional in nature. The rule discovery algorithm in Safarii discovers (propositional and multi-relational) rules with a fixed target attribute. The antecedent in this case consists of a conjunction of positive or negative conditions on the 18 remaining attributes. Rules are selected on the basis of their novelty (a.k.a. weighted relative accuracy). The rules are however of limited use to classify as yet unlabelled genes. The results of our first Data Mining approach appear in the next section.

Our next approach is to involve in the previous mining exercise more detailed information about the functional hierarchy. We continue with our 19 potential targets, but also include tables **funcat** and **hasfunction**, making our analysis multi-relational. Apart from some trivial results due to the double appearance of level 1 information, we now also expect more specific functions, up to level 5 of the hierarchy to appear in the antecedent of the rules.

Finally, we add all the available data to do a full multi-relational analysis. The aim is to find predictive models, rather than associations. Additional declarative bias was specified to ignore functional information of the gene at hand, but to include this information for any other genes related to the current gene through **homology** or **ssea**. Direct functional information in the gene table is ignored as this information will not be available either when predicting functions of as yet unseen genes.

5 Results & Interpretation

We start by showing some results for our first approach, rule discovery in the propositional data of level 1 functions. Because of the relatively small size of the table involved (4053 records by 20 attributes), the results per selected target function could be obtained very quickly, within seconds. Within the returned rules, negative rules, expressing an inverse relationship between functions, were common. This is not surprising given the non-overlapping definition of classes in FunCat. Below we give some examples of positive rules as well as a mixed rule, additionally we provide some biological justification. The numbers indicate the probability of the antecedent, the

conditional probability of the consequent, and the prior of the consequent, respectively.

TRANSCRIPTION (11) ∧ CELLULAR TRANSPORT, TRANSPORT FACILITATION AND TRANSPORT ROUTES (20) →PROTEIN WITH BINDING FUNCTION OR COFACTOR REQUIREMENT (16) (1.5%, 53.2%, 22.0%)

This rule makes sense from a biological point of view. If a protein is involved in transcription and cellular transport, common sense would say that this specific protein transports a molecule involved in transcription, e.g. RNA. The target, being category 16, tells us that this specific protein has binding qualities, which tells us that it binds to a transcription-associated factor and transports this factor to a specific place. A good example for this rule is protein ytypl190c, which belongs to the categories 11.02.03.04 *transcriptional control*, 11.04.03 *mRNA processing*, 16.03.03 *RNA binding* and 20.01.21 *RNA transport*. So from the terms *transcriptional control*, *mRNA processing* and *RNA transport*, it can be inferred that the protein also has binding qualities, in this case being the binding of an RNA molecule.

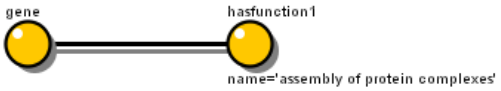
INTERACTION WITH THE CELLULAR ENVIRONMENT (34) ∧ CELLULAR COMMUNICATION/SIGNAL TRANSDUCTION MECHANISM (30) →PROTEIN ACTIVITY REGULATION (18) (2.1%, 36.8%, 5.1%)

CELLULAR COMMUNICATION/SIGNAL TRANSDUCTION MECHANISM (30) ∧ ¬ CELLULAR TRANSPORT, TRANSPORT FACILITATION AND TRANSPORT ROUTES (20) →PROTEIN ACTIVITY REGULATION (18) (4.4%, 28.3%, 5.1%)

CELLULAR COMMUNICATION/SIGNAL TRANSDUCTION MECHANISM (30) ∧ CELL RESCUE, DEFENSE AND VIRULENCE (32) →INTERACTION WITH THE ENVIRONMENT (36) (1.2%, 4.1%, 0.1%)

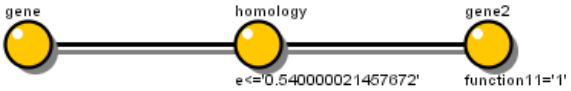
Cell rescue often has something to do with response to extra-cellular influences, like high pH, high salt concentrations etc. So it is clear that it has to somehow interact with the environment to initiate cell rescue. The protein with identifier ytylr362w is a good example. It belongs to the categories 30.01.05.01.03 *MAPKK cascade*, 30.05 *transmembrane signal transduction*, 32.01.03 *osmotic and salt stress response*, 32.01.11 *nutrient starvation response* and 36.20.35.09.05 *osmotic response*. The MAPKK cascade is also known to transduce signals in response to a variety of growth factors, cytokines and stress, so it fits perfectly in this picture. One could say that this specific protein responds to salt stress via a transmembrane-MAPKK-cascade and initiates the response to this stress.

We now turn to our second approach: finding multi-relational associations that include specific functions. The following is an example of an antecedent in a rule with target 16 *PROTEIN WITH BINDING FUNCTION OR COFACTOR REQUIREMENT*. There clearly is an association with function 14.11 *assembly of protein complexes*, which can compete with the associations mentioned above.



(4.3%, 57.9%, 22%)

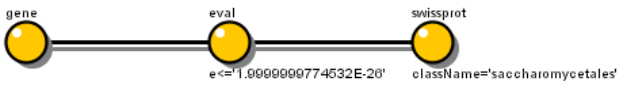
The following two rules (again target 16) predict the function of new genes based on their homology with other genes that have functions 16 and 11 (*TRANSCRIPTION*), respectively.



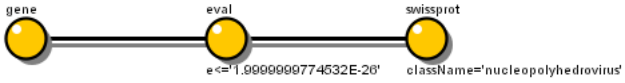
(56.6%, 29.0%, 22%)

This rule predicts the following: Given a protein *A* with function 11 and a similar protein *B* with *e*-value of this similarity below 0.54, then *B* is predicted to have function 16. What this rule says is that proteins with function 11 and proteins with function 16 are closely related. Proteins involved in transcription, like proteins dealing with RNA synthesis, RNA processing or RNA modification, often also have binding functions. Although some proteins with function 11 do not have function 16, they still could have one domain which is responsible for binding a specific biological entity (e.g. another protein or RNA), hence the low *e*-value in a homology search.

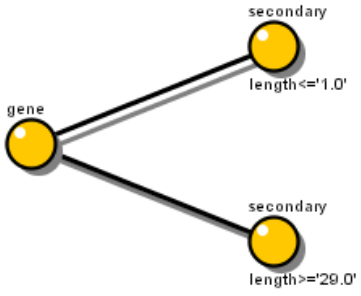
The following rules were obtained for target 36 *INTERACTION WITH THE ENVIRONMENT*:



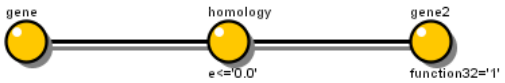
(3.1%, 1.6%, 0.1%)



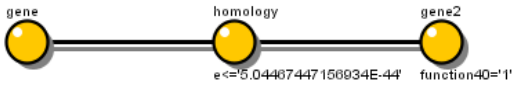
(1.1%, 4.4%, 0.1%)



(52.3%, 0.2%, 0.1%)



(4.5%, 1.1%, 0.1%)



(2.3%, 2.2%, 0.1%)

The first rule says that if a protein is very similar to another protein, with an *e*-value lower than 2e-26, and this second protein comes from the organism *saccharomycetales*, which is derived from the SwissProt-entry, then the original protein belongs to the functional category 36. This should not come as a surprise, because *Saccharomyces cerevisiae* belongs to the class *Saccharomycetaceae*, which is a subclass of *Saccharomycetales*. So the main reason for this rule is that through the course of evolution, proteins in this category remained intact, so this might be a case of true homology of proteins due to a common ancestor.

Interestingly, this is true for another organism, the *nucleopolyhedrovirus* which belongs to the *Baculoviridae*. This could mean that the proteins involved are conserved by evolution, although the virus and the yeast might not have a common ancestor. Another possible explanation would be that this similarity occurs between proteins which take part in interaction with the environment and these proteins, regardless of the organism they occur in, share a similar sequence and structure.

6 Conclusion

The preliminary findings presented in this paper show that interesting results can be obtained from the yeast database. The results make sense biologically after an initial informal inspection by a domain expert. Due to constraints of space and time we were only able to perform a shallow analysis, but we expect more specific results after a more substantial analysis. Input from the data-owners on good target concept definitions would help in this respect.

References

1. Fontana, P. and Bindewald, E. and Toppo, S. and Velasco, R. and Valle, G. and Tosatto, S.C.E., *The SSEA Server for Protein Secondary Structure Alignment*, Bioinformatics 21:3, 2005
2. *ILP 2005 Challenge*, 2005, http://www.protein-logic.com/data.html
3. Knobbe, A.J., *Multi-Relational Data Mining*, Ph.D. dissertation, 2004, http://www.kiminkii.com/thesis.pdf
4. *Safarii, the Multi-Relational Data Mining engine*, Kiminkii, 2005, http://www.kiminkii.com/safari.html