Learning Relational Descriptions of Differentially Expressed Gene Groups

Igor Trajkovski, Filip Železný, Nada Lavrač, and Jakub Tolar

***Abstract*— This paper presents a method that uses gene ontologies, together with the paradigm of relational subgroup discovery, to find compactly described groups of genes differentially expressed in specific cancers. The groups are described by means of relational logic features, extracted from publicly available gene ontology information, and are straightforwardly interpretable by medical experts. We applied the proposed method to three gene expression data sets with the following respective sets of sample classes: (i) acute lymphoblastic leukemia (ALL) vs. acute myeloid leukemia (AML), (ii) seven subtypes of ALL, and (iii) fourteen different types of cancers. Significant number of discovered groups of genes had a description which highlighted the underlying biological process that is responsible for distinguishing one class from the other classes. The quality of the discovered descriptions was also verified by crossvalidation. We believe that the presented approach will significantly contribute to the application of relational machine learning to gene expression analysis, given the expected increase in both the quality and quantity of gene/protein annotations in the near future.**

***Index Terms*— Learning in bioinformatics, Relational learning, Learning from structured data, Inductive logic programming, Scientific discovery, Microarray data analysis**

I. Introduction

MICROARRAYS are at the center of a revolution in biotechnology, allowing researchers to simultaneously monitor the expression of tens of thousands of genes. Independent of the platform and the analysis methods used, the result of a microarray experiment is, in most cases, a list of genes found to be differentially expressed in different types of tissues. A common challenge faced by the researchers is to translate such gene lists into a better understanding of the underlying biological phenomena.

Manual or semi-automated analysis of large-scale biological data sets typically requires biological experts with vast knowledge of many genes, to decipher the known biology accounting for genes with correlated experimental patterns. The goal is to identify the relevant "functions", or the global cellular activities, at work in the experiment. For example, experts routinely scan gene expression clusters to see if any of the clusters are explained by a known biological function. Efficient interpretation of this data is challenging because the number and diversity of genes exceed the ability of any single researcher to track the complex relationships hidden in the data sets. However, much of the information relevant to the data is contained in the publicly available gene ontologies and annotations. Including this additional data as a direct knowledge source for any algorithmic strategy may greatly facilitate the analysis.

We present a method to identify groups of differentially expressed genes that have functional similarity in the background knowledge formally represented with gene annotation terms from the gene ontology. The input to our algorithm is a multi-dimensional numerical data set, representing the expression of the genes under different conditions (that define the classes of examples), and an ontology used for producing background knowledge about these genes. The output is a set of gene groups whose expression is significantly different for one class compared to the other classes. The features describe the differentially expressed genes in terms of their functionality and interactions with other genes. Medical experts are usually not satisfied with a separate description of every important gene, but want to know the processes that are controlled by these genes. With our algorithm we are able to find these processes and the cellular components where they are "executed", indicating the genes from the preselected list of differentially expressed genes which are included in these processes.

These goals can be achieved by using the methodology of Relational Subgroup Discovery (RSD) [1]. With RSD we were able to induce sets of rules characterizing the differentially expressed genes in terms of functional knowledge extracted from the gene ontology and information about gene interactions.

The paper is organized as follows. In Section II we give background information about the microarray technology and gene expression analysis. Section III presents the fundamental idea of our approach, and the steps taken in our analysis. Section IV provides details of the RSD algorithm. Section V presents the results of the experiments. In Section VI we compare our approach with existing methodologies and draw some conclusions of our work.

II. Background

A. Measuring gene expression

The process of transcribing a gene's DNA sequence into the RNA that serves as a template for protein production is known as *gene expression*. A gene's expression level indicates an approximate number of copies of the gene's RNA produced in a cell. This is considered to be correlated with the amount of corresponding protein made.

Igor Trajkovski is with the Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia, email igor.trajkovski@ijs.si, Filip Železný is with the Czech Technical University, Technická 2, Prague, Czech Republic, email zelezny@fel.cvut.cz, Nada Lavrač is with the Jožef Stefan Institute, Jamova 39, Ljubljana and the University of Nova Gorica, Slovenia, email nada.lavrac@ijs.si and Jakub Tolar is with the University of Minnesota Division of Hematology-Oncology and Blood and Marrow Transplantation, USA, email tolar003@umn.edu

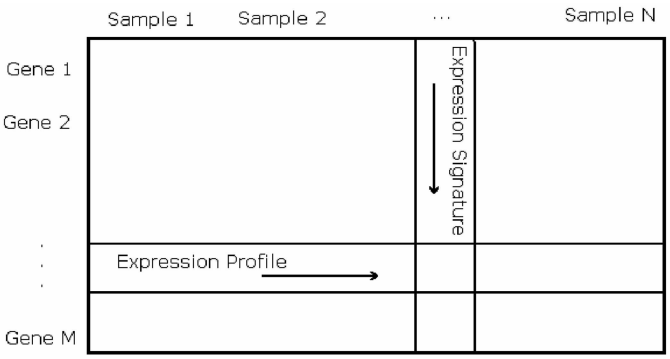

Fig. 1. The outcome of a microarray experiment is a gene expression matrix that is a 2D matrix containing the expressions of genes per sample.

While the traditional technique for measuring gene expression is labor-intensive and produces an approximate quantitative measure of expression, new technologies have greatly improved the resolution and the scalability of gene expression monitoring. *Expression chips (DNA chips, microarrays)*, manufactured using technologies derived from computer-chip production, can now measure the expression of thousands of genes simultaneously, under different conditions. These conditions may be different time points during a biological process, such as the yeast cell cycle or drosophila development; direct genetic manipulations on a population of cells such as gene deletions; or they can be different tissue samples with some common phenotype (such as different cancer specimens). A typical gene expression data set is a matrix, with each row representing a gene and each column representing a class labeled sample, e.g. a patient diagnosed having a specific sort of cancer. The value at each position in the matrix represents the expression of a gene for the given sample (see Figure 1).

B. Analysis of gene expression data

Large scale gene expression data sets include thousands of genes measured at dozens of conditions. The number and diversity of genes make manual analysis difficult and automatic analysis methods necessary. Initial efforts to analyze these data sets began with the application of unsupervised machine learning, or clustering, to group genes according to the similarity in gene expression [2]. Clustering allows for easier manual examination of the data. In typical studies, researchers examine the clusters to find those containing genes with common biological properties, such as the common molecular function or involvement in the same biological processes. After commonalities have been identified (often manually) it becomes possible to understand the global aspects of the biological phenomena studied. As the community developed interest in this area, additional novel clustering methods were introduced and evaluated for gene expression data [3], [4].

The analysis of microarray gene expression data for various tissue samples has enabled researchers to determine gene expression profiles characteristic of the disease subtypes. The groups of genes involved in these genetic profiles are rather large and a deeper understanding of the functional distinction between the disease subtypes might help not only to select highly accurate "genetic signatures" of the various subtypes, but hopefully also to select potential targets for drug design. Most current approaches to microarray data analysis use (supervised or unsupervised) machine learning algorithms to deal with numerical expression data. While clustering methods provide some insight into the data, they do not identify the critical background biological information the researcher can use to understand the significance of each cluster. However, biological knowledge in terms of functional annotations of genes is already available in public databases. Direct inclusion of this knowledge source can greatly improve the analysis, support (in term of user confidence) and explain the obtained numerical results.

C. Gene ontologies

One of the most important tools for the representation and processing of information about gene products and functions is the Gene Ontology (GO)[1]. GO is being developed in parallel with the work on a variety of other biological databases within the umbrella project Open Biological Ontologies (OBO)[2]. It provides a controlled vocabulary for the description of cellular components, molecular functions, and biological processes.

As of June 2006, GO contains 1696 cellular component, 7429 molecular function and 10668 biological process terms. Terms are organized in parent-child hierarchies (see Figure 2), indicating either that one term is more specific than another (is_a) or that the entity denoted by one term is part of the entity denoted by another (part_of). Typically, such associations (or "annotations") are first of all established by automated means and later validated by a process of manual verification which requires the annotator to have expertise both in the biology of the genes and gene products and in the structure and content of GO. The Gene Ontology, in spite of its name, is not an ontology in the sense accepted by computer scientists, in that it does not deal with axioms and formalized definitions associated to terms. It is rather a taxonomy, or, as the GO Consortium puts it, a "controlled vocabulary" providing a practically useful framework for keeping track of the biological annotations applied to gene products.

Recently, an automatic ontological analysis approach using GO has been proposed to help in solving the task of interpreting the results of gene expression data analysis [5]. From 2003 to 2005, 13 other tools have been proposed for this type of analysis and more tools continue to appear daily. Although these tools use the same general approach, identifying statistically significant GO terms that cover a selected list of genes, they differ greatly in many respects that influence in an essential way the results of the analysis. A general idea and comparison of these tools is presented in [6]. Another approach to descriptive analysis of gene expression data is [7], where a method is presented that uses text analysis to help find meaningful gene expression patterns that correlate with the underlying biology as described in the scientific literature.

[1] http://www.geneontology.org
[2] http://obo.sourceforge.net

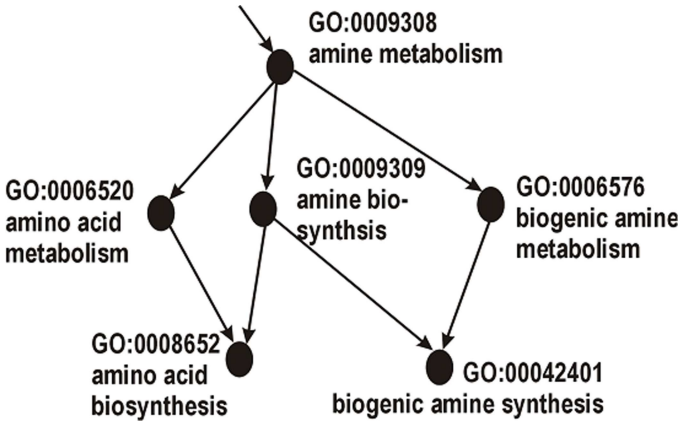

Fig. 2. The Gene Ontology provides a controlled vocabulary to describe gene product attributes in any organism.

D. Relational logic analysis and related work

While the GO based tools reviewed above enable basic analysis such as identifying a set of statistically over-represented GO terms associated with a given gene set, such analysis may be insufficient to discover frequent yet more complex ontological patterns. For example, a set of differentially expressed genes may be better characterized in terms of a logical conjunction/disjunction of GO terms presence/absence statements, rather than a simple list of frequent terms. More generally, one should also take into account the GO terms associated not only to the analyzed gene set, but also to other genes that interact with some of the analyzed genes.

The formalism of relational logic used by the RSD algorithm can capture such patterns [1], [8]. Paper [9] is related to our work in that it also uses relational logic descriptions for functional discrimination of genes. A principal difference from our approach is however at least threefold. Firstly, [9] uses the inductive logic programming system Progol to search for relational ontological patterns (rules). The cover-set algorithm used by Progol is arguably inappropriate for finding a set of interesting gene subgroup descriptions as we explain later in the paper. On the contrary, our approach is based on the weighted covering algorithm more suitable for such a defined task. Secondly and more importantly, the approach in [9] assumes all genes in the analyzed gene set to be of the same importance when forming the pattern descriptions. This clearly ignores the fact that certain genes are more "interesting" than others, e.g. their expression variance across different conditions is larger. When constructing gene group descriptions, our approach deliberately devotes more attention to the "more important" genes than to those less important. Lastly, unlike our work, [9] does not consider interactions among genes or their inclusion in gene regulatory pathways as relational properties exploitable for descriptive purposes.

Another recent paper [10] also uses relational logic for learning from genomic, proteomic and related data sources, including gene ontologies. The learning objective of [10] is however rather unrelated to ours. Whereas we attempt to compactly describe differentially expressed gene groups, [10] aims to predict protein-protein interactions.

III. Descriptive analysis of gene expression data

The fundamental idea of this paper is outlined in Figure 3. First, we construct a set of differentially expressed genes, $G_C(c)$, for every class $c \in C$. These sets can be constructed in several ways. For example: $G_C(c)$ can be the set of k $(k > 0)$ most correlated genes with class c, for instance computed by Pearson's correlation. $G_C(c)$ can also be the set of best k single gene predictors, using the recall values from a microarray experiment (absent/present/marginal) as the expression value of the gene. These predictors can acquire the form such as:

$$\text{If } gene_i = \textit{present} \text{ Then } \textit{class} = c$$

In our experiments $G_C(c)$ was constructed using a modified version of the t-test statistics. Details about the selection mechanism used in our experiments are presented in Section V.

The second step aims at improving the interpretability of G_C. Informally, we do this by identifying subgroups of genes in $G_C(c)$ (for each $c \in C$) which can be summarized in a compact way. Put differently, for each $c_i \in C$ we search for compact descriptions of gene subgroups with expression strongly correlating (positively or negatively) with c_i and weakly with all $c_j \in C, j \neq i$.

Searching for these groups of genes, together with their description, is defined as a separate supervised machine learning task. We refer to it as the secondary mining task, as it aims to mine from the outputs of the primary learning process in which differentially expressed genes are searched. This secondary task is, in a way, orthogonal to the primary discovery process in that the original attributes (genes) now become training examples, each of which has a class label "differentially expressed" and "not differentially expressed". To apply a discovery algorithm, information about relevant features of these examples are required. No such features (i.e., "attributes" of the original attributes - genes) are usually present in the gene expression microarray data sets themselves. However, this information can be extracted from a public database of gene annotations (we used the Entrez Gene database[3] maintained at the US National Center for Biotechnology Information). For each gene we extracted its molecular functions, biological processes and cellular components where its protein products are located, and transformed this information into the gene's *background knowledge* encoded in relational logic in the form of Prolog facts. Part of the knowledge for gene SRC, whose Entrez GeneID is 6714, is presented here:

```
function(6714,'ATP binding').
function(6714,'receptor activity').
process(6714,'signal complex formation').
process(6714,'protein kinase cascade').
component(6714,'integral to membrane').
...
```

Next, using GO, in the gene's background knowledge we also included the gene's generalized annotations. For example, if one gene is functionally annotated as: "zinc ion binding", in the background knowledge we also included its more general functional annotations: `transition metal ion binding`, `metal ion binding`, `cation`

[3] ftp://ftp.ncbi.nlm.nih.gov/gene/

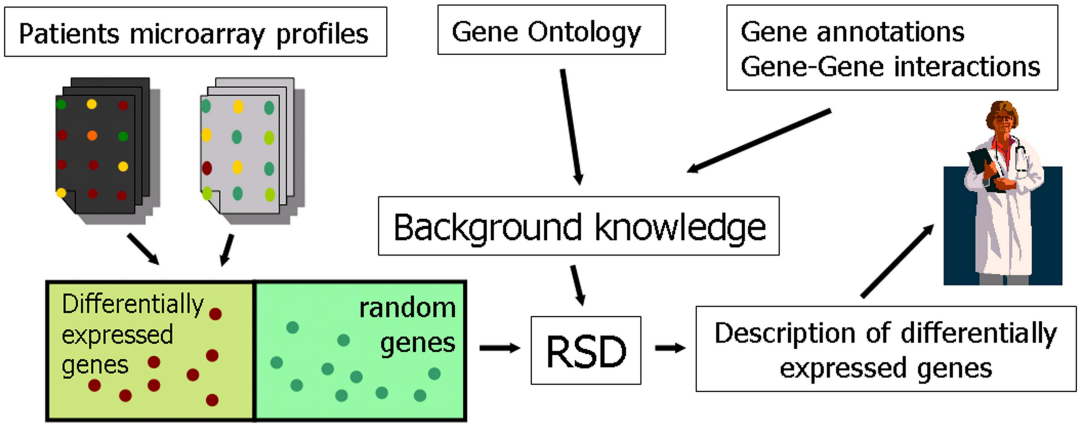

Fig. 3. An outline of the process of microarray data analysis using RSD. First, in microarray data, we search for differentially expressed genes. Using the gene ontology information, gene annotation and gene interaction data, we produce background knowledge for differentially expressed genes on one hand, and randomly chosen genes on the other hand. The background knowledge is represented in the form of Prolog facts. Next, the RSD algorithm finds characteristic descriptions of the differentially expressed genes. Finally, the discovered descriptions can be straightforwardly interpreted and exploited by medical experts.

`binding`, `ion binding` and `binding`. In the gene's background knowledge we also included information about the interactions of the genes, in the form of pairs of genes for which there is an evidence that they can interact:

```
interaction(6714,155).
interaction(6714,1874).
interaction(6714,8751).
interaction(6714,302).
...
```

In traditional machine learning, examples are expected to be described by a tuple of values corresponding to some predefined, fixed set of attributes. Note that a gene annotation does not straightforwardly correspond to a fixed attribute set, as it has an inherently relational character and we need to develop the relevant attributes on the basis of the pre-formed relational background knowledge. For example, a gene may be related to a variable number of cell processes, meaning it can play a role in a variable number of regulatory pathways etc. This imposes 1-to-many relations hard to elegantly capture within an attribute set of a fixed size. Furthermore, a useful piece of information about a gene g may, for instance, be expressed by the following feature involving the background knowledge of another gene:

gene g interacts with another gene whose functions include protein binding. (*)

Going even further, the feature may not include only a single interaction relation but rather consider entire chains of interactions. Consequently, the task we are approaching is a case of *subgroup discovery from relational data*. For this purpose we employ the methodology of relational subgroup discovery proposed in [1], [8] and implemented in the RSD[4] algorithm. Using RSD, we were able to discover knowledge such as:

Genes whose protein products are located in the nucleus, interacting with genes involved in the process of transcription regulation tend to be differentially expressed between acute myeloid leukemia and acute lymphoblastic leukemia.

[4]http://labe.felk.cvut.cz/~zelezny/rsd/rsd.pdf

IV. The RSD Algorithm

The RSD algorithm proceeds in two steps. First, it constructs a set of relational features in the form of first-order logic atom conjunctions. The entire set of features is then viewed as an attribute set, where an attribute has the value *true* for a gene (example) if the gene has the feature corresponding to the attribute. As a result, by means of relational feature construction we achieve the conversion of relational data into attribute-value descriptions.[5] In the second step, interesting gene subgroups are searched, such that each subgroup is represented as a conjunction of selected features. The subgroup discovery algorithm employed in this second step is an adaptation of the popular propositional rule learning algorithm CN2 [13].

A. Relational feature construction

The feature construction component of RSD aims at generating a set of relational features in the form of relational logic atom conjunctions. For example, the feature (*) exemplified informally in the previous section has the relational logic form:

```
interaction(A,B),function(B,'protein binding')
```

where upper cases denote variables, and a comma between two logical literals denotes a conjunction.

The user specifies *mode declarations* which syntactically constrain the resulting set of constructed features. Each mode declaration defines a predicate that can appear in a feature, and assigns to each of its arguments a *type* and a *mode* (either input or output). Thus the following example declaration

```
mode(3, interaction(+gene,-gene))
```

states that predicate `interaction` can appear in the feature with an input (+ sign) variable of type `gene` and an output (- sign) variable of the same type. The first declaration argument (number 3) stipulates that the predicate can appear in a single feature at most 3 times with the same input variable; in other

[5]This process is known as *propositionalization* [11],[12].

words, three interactants of a single gene can be addressed in a feature.

In a feature, if two arguments have different types, they may not hold the same variable. Also, literals in a feature must be "linked":

1) Every variable in an input argument of a literal must appear in an output argument of some preceding literal in the same feature, with the exception of the first variable in the feature (the *key* variable).
2) Inversely, every output variable of a literal must appear as an input variable of some subsequent literal.

Furthermore, the maximum length of a feature (number of contained literals) is declared, along with further optional syntactic constraints [1], [8].

Predicates with only variables in their arguments are not sufficient to capture important gene's properties. It is important that features may also contain constants (such as `'protein binding'`). A distinguished predicate `instantiate` is used to indicate variables which will be automatically substituted by constants used in the training examples. For example, with the following declaration

```
mode(2, function(+gene,-function))
mode(1, instantiate(+function))
```

RSD first generates a constant-free feature

```
interaction(A,B), function(B,C), instantiate(C)
```

and then replaces it with a *set* of features, in each of which variable `C` is replaced by a constant and the `instantiate` predicate is removed. An example feature set consists of the following two features:

```
interaction(A,B), function(B,'protein binding')
```

and

```
interaction(A,B), function(B,'binding')
```

However, only such replacements for `C` are considered that make the resulting feature hold true for at least a pre-specified number of genes, according to a pre-specified minimal support threshold of RSD.

Given a set of declarations, RSD proceeds in the manner described above to produce an exhaustive set of features satisfying the declarations. Technically, this is implemented as an exhaustive depth-first backtrack search in the space of all feature descriptions, equipped with certain pruning mechanisms. Besides the language declarations, each feature must also comply to the *connectivity* requirement, according to which no feature may be decomposable into a conjunction of two or more features. For example, the following expression does not form an admissible feature:

```
interaction(A,B),function(B,'protein binding'),
interaction(A,C), component(C,'membrane')
```

The reason is that it can be decomposed into two separate features, consisting of the first two (last two, respectively) literals. We do not construct such decomposable expressions, as these are clearly redundant for the purpose of subsequent search for rules with conjunctive antecedents. Note that decomposable features may in general be made undecomposable by adding a literal, such as by adding `interaction(B,C)` to the expression exemplified above. It is primarily the concept of undecomposability that allows for extensive search space pruning [1], [8] in the feature construction process.

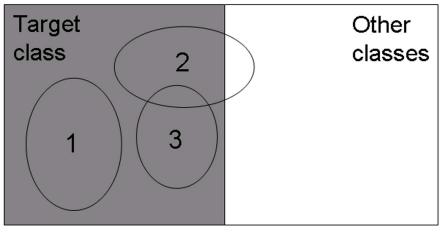

Fig. 4. Descriptions of discovered subgroups ideally cover just individuals of the target class (subgroups 1 and 3), however they may cover also a few individuals of other classes (subgroup 2).

Some examples of features constructed by RSD are listed below:

```
f(7,A):-function(A,'kisspeptin rec. binding').
f(8,A):-function(A,'phosphopant. binding').
f(11,A):-process(A,'intestinal lipid catabol').
f(14,A):-process(A,'neurite morphogenesis').
f(19,A):-component(A,'nucleus').
f(22,A):-interaction(A,B),
         function(B,'mannokinase activity').
f(24,A):-interaction(A,B),
         function(B,'enzyme regulator act.'),
         component(B,'membrane').
f(84,A):-interaction(A,B),
         process(A,'glycolate catabolism'),
         component(B,'intrinsic to membrane').
```

where the "head" of the feature definition formally indicates the feature number and the key variable.

Finally, to evaluate the truth value of each feature for each example for generating the attribute-value representation of the relational data, the first-order logic resolution procedure is used, provided by a standard Prolog language interpreter.

B. Subgroup Discovery

Subgroup discovery aims at finding population subgroups that are statistically "most interesting", e.g., are as large as possible and have the most unusual statistical characteristics with respect to the property of interest [14] (see Figure 4).

Notice an important aspect of the above definition: there is a predefined property of interest, meaning that a subgroup discovery task aims at characterizing population subgroups of a given *target* class. This property indicates that standard classification rule learning algorithms could be used for solving the task. However, while the goal of classification rule learning is to generate predictive models in the form of rule sets that discriminate between the target class and non-target classes, subgroup discovery aims at discovering a set of individual patterns (rules) characterizing the target class.

Rule learning typically involves two main procedures: the search procedure that performs a search to find a single rule (see Subsection 1 below) and the control procedure (the covering algorithm) that repeatedly executes the search in order to induce a set of rules (see Subsection 2).

1) Inducing a single subgroup describing rule: Our algorithm RSD [1], [8] is based on an adaptation of the standard

propositional rule learner CN2 [13]. Its search procedure used in learning a single rule performs beam search, starting from the empty conjunct, successively adding conditions (relational features). In CN2, classification accuracy of a rule is used as a heuristic function in the beam search. The accuracy[6] of an induced rule of the form $H \leftarrow B$ (where H in the rule head is the target class, and B is the rule body formed of a conjunction of relational features) is equal to the conditional probability of head H, given that body B is satisfied: $p(H|B)$.

In RSD, the accuracy heuristic $Acc(H \leftarrow B) = p(H|B)$ is replaced by the *weighted relative accuracy* heuristic. Weighted relative accuracy is a reformulation of one of the heuristics used in MIDOS [14] aimed at balancing the size of a group with its distributional unusualness [15]. It is defined as follows:

$$WRAcc(H \leftarrow B) = p(B) \cdot (p(H|B) - p(H)). \quad (1)$$

Weighted relative accuracy consists of two components: generality $p(B)$, and relative accuracy $p(H|B) - p(H)$. The second term, relative accuracy, is the accuracy gain relative to fixed rule $H \leftarrow true$. The latter rule predicts all instances to satisfy H; a rule is only interesting if it improves upon this "default" accuracy. Another way of viewing relative accuracy is that it measures the utility of connecting rule body B with rule head H. Note that it is easy to obtain high relative accuracy with very specific rules, i.e., rules with low generality $p(B)$. To this end, generality is used as a "weight" which trades off generality of the rule (rule coverage $p(B)$) and relative accuracy ($p(H|B) - p(H)$).

In the computation of *Acc* and *WRAcc* all probabilities are estimated by relative frequencies[7] as follows:

$$Acc(H \leftarrow B) = p(H|B) = \frac{p(HB)}{p(B)} = \frac{n(HB)}{n(B)} \quad (2)$$

$$WRAcc(H \leftarrow B) = \frac{n(B)}{N}\left(\frac{n(HB)}{n(B)} - \frac{n(H)}{N}\right) \quad (3)$$

where N is the number of all the examples, $n(B)$ the number of examples covered by rule $H \leftarrow B$, $n(H)$ the number of examples of class H, and $n(HB)$ the number of examples of class H correctly classified by the rule (true positives).

2) Inducing a set of subgroup describing rules: In CN2, for a given class in the rule head, the rule with the best value of the heuristic function found in the beam search is kept. The algorithm then removes all examples of the target class satisfying the rule's conditions (i.e., positive examples *covered* by the rule) and invokes a new rule learning iteration on the remaining training set. All negative examples (i.e., examples that belong to other classes) remain in the training set.

In this classical covering algorithm, only the first few induced rules may be of interest as subgroup descriptors with sufficient coverage, since subsequently induced rules are induced from biased example subsets, i.e., subsets including only positive examples not covered by previously induced rules. This bias constrains the population of individuals in a way that is unnatural for the subgroup discovery process, which is aimed at discovering characteristic properties of subgroups of the target population.

In contrast, RSD uses the *weighted covering algorithm*, which allows for discovering interesting subgroup properties in the entire target population. The weighted covering algorithm modifies the classical covering algorithm in such a way that covered positive examples are not deleted from the set of examples to be used to construct the next rule. Instead, in each run of the covering loop, the algorithm stores with each example a count that indicates how many times (with how many induced rules) the example has been covered so far.

By default, initial weights of all examples e_j are set to 1 (alternatively, as was the case in our experiments, the initial weights of the examples may encode the apriori importance of a given example). In subsequent iterations of the weighted covering algorithm all target class examples weights decrease according to the formula $\frac{1}{i+1}$, where i is the number of constructed rules that cover example e_j. In this way the target class examples whose weights have not been decreased will have a greater chance to be covered in the following iterations of the weighted covering algorithm.

The combination of the weighted covering algorithm with the weighted relative accuracy thus implies the use of the following *modified WRAcc* heuristic:

$$WRAcc(H \leftarrow B) = \frac{n'(B)}{N'}\left(\frac{n'(HB)}{n'(B)} - \frac{n(H)}{N}\right) \quad (4)$$

where N is the number of examples, N' the sum of the weights of all examples, $n(H)$ the number of examples of class H, $n'(B)$ the sum of the weights of all covered examples, and $n'(HB)$ the sum of the weights of all correctly covered examples.

V. Experiments

A. Materials and methods

We apply the proposed methodology on three classification problems from gene expression data, with the aim to describe the genes that are usually used by the classifiers, the differentially expressed genes selected as the target class.

The first problem was introduced in [18] and aims at distinguishing between samples of ALL and AML from gene expression profiles obtained by the Affymetrix HU6800 microarray chip, containing probes for 6817 genes. The data contains 73 class-labeled samples of expression vectors. The second problem was described in [19] and aims at distinguishing different subtypes of ALL (6 recognized subtypes plus a separate class "other" containing the remaining samples). The data contains 132 class-labeled samples obtained by Affymetrix HG-U133 set of microarrays, containing 22283 probes. The third problem was defined in [20]. Here one tries to distinguish among 14 classes of cancers from gene expression profiles obtained by the Affymetrix Hu6800 and Hu35KsubA microarray chip, containing probes for 16,063 genes. The data set contains 198 class-labeled samples. Note that this paper does not address the learning task of discriminating between the classes. Instead, for the given target class we aim at finding the most characteristic description of its differentially expressed genes.

[6] In some contexts, this quantity is called *precision*.

[7] Alternatively, the Laplace [16] and the m-estimate [17] could also be used.

To access the annotation data for every gene considered, it was necessary to obtain unique gene identifiers from the microarray probe identifiers available in the original data. We achieved this by script-based querying of the Affymetrix site[8] for translating probe ID's into unique gene ID's. Knowing the gene identifiers, information about gene annotations and gene interactions can be extracted from the Entrez gene information database[9]. We developed a program script[10] in the Python language, which extracts gene annotations and gene interactions from this database, and produces their structured, relational logic representations which can be used as input to RSD.

For all three data sets, and for each class c we first extracted a set of differentially expressed genes $G_C(c)$. In our experiments we used t-test score $T(g,c)$ for selecting differentially expressed genes. t-test is a test of the null hypothesis that the means of two normally distributed populations are equal. Higher $|T(g,c)|$ means higher probability which in turn means that mean gene expression is different between different classes.

$T(g,c)$ is computed by the following formula:

$$T(g,c) = \frac{\mu_1(g) - \mu_2(g)}{\sqrt{\frac{\sigma_1(g)}{N_1} + \frac{\sigma_2(g)}{N_2}}} \quad (5)$$

where $N_1 = |c|$, $N_2 = |C \setminus c|$, $[\mu_1(g), \sigma_1(g)]$ and $[\mu_2(g), \sigma_2(g)]$ denote the means and standard deviations of the logarithm of the expression levels of gene g for the samples in class c and samples in $C \setminus c$, respectively.

$T(g,c)$ reflects the difference between the classes relative to the standard deviation within the classes. Large values of $|T(g,c)|$ indicate a strong correlation between the expression of gene g and class c, while the sign of $T(g,c)$ being positive (negative) corresponds to g being highly (less) expressed in class c than in the other classes. Unlike a standard Pearson's correlation coefficient, $T(g,c)$ is not confined to the range $[-1,+1]$. In order to avoid situations illustrated in Figure 5, where genes B and C would have similar values of $|T(g,c)|$ but where C is not significantly differentially expressed, we dictate one more condition for a gene to be selected: $|\mu_1(g) - \mu_2(g))| > 1$. Thereby we ensure that selected genes have at least twofold difference in their average expression for the given class.

For all three problems and all classes we selected the 50 most differentially expressed (highest t-score ranking) genes and the same number of randomly chosen non-differentially expressed genes. The specific number of selected genes is a matter of trade-off. Including a high number of examples in the training set is in general preferable for learning. However, extending the training set to relatively low-scoring genes decreases the overall quality of the training set. A full quantification of this trade-off is out of the scope of this study, where we adhere to 50 examples of each class. This is a usual number of selected genes in the context of microarray data classification with support vector machines or voting algorithms [18].

[8]www.affymetrix.com/analysis/netaffx/

[9]ftp://ftp.ncbi.nlm.nih.gov/gene/

[10]This script is available on request to the first author.

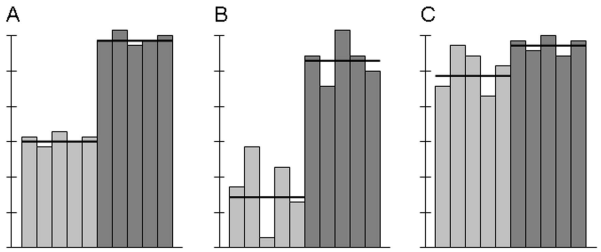

Fig. 5. Expression of three genes (A, B and C) for five patients of class 1 and five patients of class 2. Perfect class distinction can be achieved by idealized gene A, in which the expression level is uniformly low in class 1 and uniformly high in class 2. A more realistic case is gene B which is also useful for class distinction. We do not use gene C for class distinction as we are interested in genes that have significant difference in their mean expression between classes.

TABLE I

AVERAGE, MAXIMAL AND MINIMAL VALUE OF $\{|T(g,c)|, g \in G_C(c)\}$ FOR EACH PROBLEM AND CLASS c.

TASK	CLASS	AVG	MAX	MIN
ALL-AML	ALL	6.74	11.09	5.31
	AML	6.74	11.09	5.31
SUBTYPES OF ALL	BCR	5.95	10.30	4.65
	E2A	11.68	38.80	8.46
	HD50	6.09	8.56	5.21
	MLL	8.71	13.15	6.85
	T_ALL	16.70	27.12	12.66
	TEL	9.69	17.59	7.34
MULTI CLASS	BREAST	6.53	8.42	5.86
	PROSTATE	6.05	11.90	4.84
	LUNG	5.04	8.56	4.25
	COLORECTAL	5.71	14.83	4.42
	LYMPHOMA	8.73	14.69	7.32
	BLADDER	5.91	10.27	5.07
	MELANOMA	6.53	11.28	5.71
	UTERUS	5.07	7.49	4.46
	LEUKEMIA	11.55	17.02	9.78
	RENAL	4.65	6.62	4.06
	PANCREAS	5.22	7.92	4.32
	OVARY	4.06	6.33	3.59
	MESOTHELIOMA	4.81	9.51	4.61
	CNS	11.99	23.06	9.47

The average, maximal and minimal values of $|T(g,c)|$ for the selected differentially expressed genes for each problem/class are listed in Table I. In general, higher numbers mean that the class is easier to distinguish from the other classes on the basis of the gene signature.

The usage of the gene t-test score $T(g,c)$ is twofold. In the first part of the analysis it is used for the selection of differentially expressed genes as described above. Secondly, it acts as the initial weight for each example-gene in the subgroup discovery procedure where we try to characterize these differentially expressed genes. In this secondary mining task, RSD will thus prefer to group genes with large weights. As a consequence, such important genes are typically covered by more than one reported subgroup description, each time with an alternative description.

B. Example Result

To illustrate the straightforward interpretability of the induced subgroup descriptions, we use as an example the best-scoring gene subgroup discovered by RSD for the CNS (central nervous system) cancer class from the 14-class cancer problem. A group of genes, $geneGroup(A)$, differentially expressed between CNS on one hand and the other classes, was defined by RSD through the conjunction of two relational logic features:

```
interaction(A,B), process(B,'phosphorylation')
```

and

```
interaction(A,B), process(B,'negative regulation of
apoptosis'), component(B,'intracellular
membrane-bound organelle')
```

This gene group, defined by the interaction with genes involved in phosphorylation, negative regulation of apoptosis and intracellular localization, contains 7 differentially expressed genes and none of the non-differentially expressed genes used as the negative examples by the algorithm. The gene group members are brain specific genes and genes active in cellular survival. The former includes glial fibrillar astrocytic protein [GFAP, 2670] and reticulon 4 [neurite growth factor, 57142] exhibited positive expression scores as would be predicted in brain derived cancers. The latter, cell death genes caspase 4 [837] and tumor necrosis factor receptor type I associated death domain protein [TRADD, 8717] are both associated with decreased expression, also an expected finding, as lower levels of these cell death/pro-apoptotic genes are associated with uncontrolled cellular growth in malignancy and are one of the most prominent features of cancers.

Of note, these observations support the validity of our method (as they fit biological expectations based on scientific and clinical investigations unrelated to ours) and thus give credibility to findings related to the remaining genes in the subgroup, of which little is known in brain cancers. These include glycogen synthase kinase 3 beta [2932] and nuclear receptor corepressor 2 [9612]. Glycogen synthase kinase 3 beta is a master switch of multiple processes involved in cellular biology by definition exercising its regulatory effects by phosphorylation. Specifically it is critical for cell migration, proliferation (including pathological cellular proliferation in multiple human cancers) and, interestingly, it has in fact been previously reported to be functionally connected to brain protein tau [21]. To our knowledge, however, nothing is known of its role on brain tumors. The role of the nuclear receptor corepressor 2 (or silencing mediator for thyroid hormone receptor, SMRT) has been described for breast cancer, prostate cancer and in impaired response to differentiation signaling in hematopoietic cells. As their role in brain cancer is not known and based on our data their expression is indeed significantly increased in brain tumors (when compared to other malignancies) the nuclear receptor corepressor 2 and glycogen synthase kinase 3 beta represent good candidate genes for further investigations in etiology of brain cancer.

TABLE II
PRECISION, RECALL AND AUC FIGURES OF FOUND SUBGROUPS, FOR THE SET OF ALL/AML, SUBTYPES OF ALL AND MULTI-CLASS-CANCER DIFFERENTIALLY EXPRESSED GENES, OBTAINED THROUGH 5-FOLD CROSS-VALIDATION.

TASK	DATA	PRE	REC	AUC
ALL-AML	Train	100(± 0)%	16%	65%
	Test	85(± 6)%	13%	60%
SUBTYPES	Train	95(± 4)%	17%	63%
OF ALL	Test	78(± 10)%	12%	61%
MULTI	Train	94(± 6)%	14%	59%
CLASS	Test	75(± 12)%	12%	57%

C. Statistical validation

Here we present a statistical validation of the proposed methodology for discovering descriptions of differentially expressed gene groups. Specifically we wish to determine if the high descriptive capacity pertaining to the incorporation of the expressive relational logic language incurs a risk of *descriptive overfitting*, i.e., a risk of discovering subgroups whose bias toward differential expression is only due to chance.

We thus aim at measuring the discrepancy of the quality of discovered subgroups on the training data set on one hand and an independent test set on the other hand. We will do this through the standard 5-fold stratified cross-validation regime.

The specific qualities measured for each set of subgroups produced for a given class are average *precision* (PRE), *recall* (REC) and *area under ROC* (AUC) values among all subgroups in the subgroup set.

Table II[11] shows the PRE and REC values results for the three respective problem domains. Overall, the results demonstrate an acceptable decay from the training to the testing set in terms of both PRE and REC, suggesting that the discovered subgroup descriptions indeed capture the relevant gene properties. In terms of *total* coverage, in average, RSD covered more then $\frac{2}{3}$ of the preselected differentially expressed genes, while $\frac{1}{3}$ of the preselected genes were not included in any group. A possible interpretation is that they are not functionally connected with the other genes and their initial selection through the t-test was due to chance. This information can evidently be back-translated into the gene selection procedure and used as a gene selection heuristic. This approach is out of the scope of this paper but represents the next step in our future work.

The risk of descriptive overfitting suggested by the results of Table II is due to two reasons: first, the imperfections in the data and second, the high expressiveness of the relational logic language.

Concerning the first reason, the existing gene annotations databases are currently rather coarse-grained in that high-confidence classification of genes into low-level (i.e. specific) ontological classes is rarely available. A second source of input imperfectness is the fact that functions, locations and involved

[11] For the first problem we had one set of differentially expressed genes, where for the second (third) problem we had 6 (14) sets of differentially expressed genes and equal number of learning tasks, one for each class, where results of each subtask were averaged.

TABLE III
PRECISION OF DISCOVERED DIFFERENTIALLY EXPRESSED GENE GROUP DESCRIPTIONS, FOR THREE SCENARIOS WHERE PART OF THE BACKGROUND KNOWLEDGE OR GENE-WEIGHT INFORMATION WAS REMOVED.

TASK	ORIG	-INTERACTION	-GO	-WEIGHTS
ALL-AML	85(± 6)%	44(± 12)%	72(± 13)%	75(± 8)%
SUBTYPES OF ALL	78(± 10)%	52(± 13)%	74(± 16)%	71(± 12)%
MULTI CLASS	75(± 12)%	45(± 16)%	56(± 14)%	73(± 14)%

processed are known for only a subset of genes. Furthermore, most annotation databases are built by curators who manually review the existing literature. It is thus possible that certain known facts get temporarily overlooked. For instance, [6] found references in literature published in the early 90s, for 65 functional annotations that are not yet included in the current functional annotation databases.

Secondly, the language expressivity allows for forming rather complex rules, involving both gene-ontological terms and gene-interaction relations. As such they are possibly prone to capturing noise in data rather than genuine biological principles.

Despite the two described factors, the overfitting effect manifests itself to an acceptable extent and the rule quality measured on independent testing sets is still relatively high. Moreover, some of the actual discovered patterns also lead to biologically plausible interpretations as demonstrated in Section V-B.

D. Analyzing Individual Components of the Methodology

We further experimented with different settings of our algorithm in order to investigate the influence of different ingredients of the approach on the precision of the found descriptions. In addition to the original setting (ORIG), we performed experiments with three alternative settings: without gene-interaction information, without GO term generalization, and without incorporating gene t-test scores as the initial weights in the RSD's weighted covering algorithm for subgroup discovery (thus initializing all weights to 1). In Table III we present the test-set results averaged in 5-fold crossvalidation. Table III shows that all the three ingredients exhibit a strong positive influence on the results, with interaction data being the strongest factor.

VI. DISCUSSION

In this paper we presented a method that uses gene ontologies, together with the paradigm of relational subgroup discovery, to help find patterns of expression for genes with a common biological function that correlate with the underlying biology responsible for class differentiation. Our methodology proposes to first select a set of important differentially expressed genes for all classes and then find compact, relational descriptions of subgroups among these genes.

It is noteworthy that the latter descriptive "post-processing" step is a machine learning task, in which the curse of dimensionality usually ascribed to microarray data classification, actually turns into an advantage. This is because, in traditional microarray data mining configurations, the high number of genes results in a high number of attributes usually confronted with a relatively small number of expression samples, thus forming grounds for overfitting. In our approach, on the contrary, genes correspond to examples and thus their abundance is beneficial. Furthermore, the dimensionality of the secondary attributes (relational features of genes extracted from gene annotations) can be conveniently controlled via suitable constraints of the language grammar used for the automatic construction of the gene features.

A further remark concerns the fact that genes are frequently associated to multiple functions, i.e. they may under some conditions exhibit a behavior of genes with one function while in other conditions a different aspect of their function may be important. Here the subgroup discovery methodology is effective at selecting a specific function important for the classification. Indeed, one given gene can be included in multiple subgroup descriptions (this was e.g. the case of genes with id's 51592 and 115426 in the breast cancer class), each emphasizing the different biological process critical to the explanation of the underlying biology responsible for observed experimental results.

Yet another aspect of the proposed method is of interest, following from the illustrative example of a discovery result provided in Section V-B. Here the discovered subgroup contains four genes whose differential expression (for the CNS cancer class) is well in accordance with the biological state of the art. The group is described using the *features shared by the genes*, rather than through plain gene list as in traditional approaches. As a consequence, the group also includes further genes sharing the features, whose connection to brain cancer has not yet been described, yet closer analysis reveals evidence that such association is indeed plausible. We believe that this "generalization" aspect of the proposed methodology may contribute to discovering new marker genes by proposing candidate genes for further experimental evaluation.

We have assessed the quality of the induced descriptions by evaluating them on independent test sets using 5-fold crossvalidation. The results show a clear advantage of using all the complementary sources of background knowledge in the description generation procedure (GO ontology, gene interactions as well as degree of differential expression of genes represented by gene weights), as shown in Table III.

We believe that the presented approach can significantly contribute to the application of relational machine learning to gene expression analysis. Despite the demonstrated benefits of the methodology, the precision and recall evaluation of descriptors in Table II suggests that there is still room for improvement. This is to be achieved through the expected increase in both the quality and quantity of gene/protein annotations in the near future.

ACKNOWLEDGMENT

Igor Trajkovski and Nada Lavrač are supported by the Slovenian Ministry of Higher Education, Science and Technology. Filip Železn´y is supported by the Czech Academy of Sciences through the project KJB201210501 Logic Based Machine Learning for Analysis of Genomic Data. Jakub Tolar is supported by the Children's Cancer Research Fund, University of Minnesota Cancer Center and Department of Pediatrics.

REFERENCES

[1] N. Lavrač, F. Železný, and P. Flach, "RSD: Relational subgroup discovery through first-order feature construction," in *Proceedings of the 12th International Conference on Inductive Logic Programming*, 2002, pp. 149–165.

[2] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," in *Proceedings of National Academy of Science USA, 95:25*, 1998, pp. 14 863–14 868.

[3] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *Journal of Computational Biology*, pp. 281–297, 1999.

[4] L. J. Heyer, S. Kruglyak, and S. Yooseph, "Exploring expression data: Identification and analysis of coexpressed genes," *Genome Research*, pp. 1106–1115, 1999.

[5] P. Khatri, S. Draghici, G. C. Ostermeier, and S. A. Krawetz, "Profiling gene expression using onto-express," *Genomics*, pp. 266–270, 2002.

[6] P. Khatri and S. Draghici, "Ontological analysis of gene expression data: current tools, limitations, and open problems," *Bioinformatics*, pp. 3587–3595, 2005.

[7] S. Raychaudhuri, H. Schutze, and R. B. Altman, "Inclusion of textual documentation in the analysis of multidimensional data sets: application to gene expression data," *Machine Learning*, pp. 119–145, 2003.

[8] F. Železný and N. Lavrač, "Propositionalization-based relational subgroup discovery with RSD," *Machine Learning*, pp. 33–63, 2006.

[9] L. Badea, "Functional discrimination of gene expression patterns in terms of the gene ontology," in *Proceedings of the Pacific Symposium on Biocomputing*, vol. 8, 2003, pp. 565–576.

[10] T. N. Tran, K. Satou, and T. B. Ho, "Using inductive logic programming for predicting protein-protein interactions from multiple genomic data," in *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2005, pp. 321–330.

[11] S. Kramer, N. Lavrač, and P. Flach, "Propositionalization approaches to relational data mining," in *Relational Data Mining*, S. Džeroski and N. Lavrač, Eds. Springer, 2001, pp. 262–291.

[12] N. Lavrač and P. Flach, "An extended transformation approach to inductive logic programming," *ACM Transactions on Computational Logic*, pp. 458–494, 2001.

[13] P. Clark and T. Niblett, "The CN2 induction algorithm," *Machine Learning*, pp. 261–283, 1989.

[14] S. Wrobel, "An algorithm for multi-relational discovery of subgroups," in *Proceedings of the First European Symposion on Principles of Data Mining and Knowledge Discovery*, 1997, pp. 78–87.

[15] W. Kloesgen, "Explora: A multipattern and multistrategy discovery assistant," in *Advances in Knowledge Discovery and Data Mining, AAAI Press*, 1996, pp. 249–271.

[16] P. Clark and R. Boswell, "Rule induction with CN2: Some recent improvements," in *Proceedings of the 5th European Working Session on Learning*, 1991, pp. 151–163.

[17] B. Cestnik, "Estimating probabilities: A crucial task in machine learning," in *Proceedings of the 9th European Conference on Artificial Intelligence*, 1990, pp. 147–149.

[18] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, pp. 531–537, 1999.

[19] M. E. Ross, X. Zhou, G. Song, S. A. Shurtleff, K. Girtman, W. K. Williams, H.-C. Liu, R. Mahfouz, S. C. Raimondi, N. Lenny, A. Patel, and J. R. Downing, "Classification of pediatric acute lymphoblastic leukemia by gene expression profile," *BLOOD*, pp. 2951–2959, 2003.

[20] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.-H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub, "Multiclass cancer diagnosis using tumor gene expression signatures," in *Proceedings of the National Academy of Sciences*, 2001, pp. 15 149–15 154.

[21] Z. Yuan, A. Agarwal-Mawal, and H. K. Paudel, "14-3-3 binds to and mediates phosphorylation of microtubule-associated tau protein by ser9-phosphorylated glycogen synthase kinase 3beta in the brain," *Journal of Biological Chemistry*, vol. 279, pp. 26 105–26 114, 2004.

Igor Trajkovski is a researcher at the Department of Knowledge Technologies of the Jožef Stefan Institute in Ljubljana, Slovenia. He received a MSc in computer science from the Saarland University and Max Plank Institute for Informatics in Germany and is now a PhD student at the Jožef Stefan International Postgraduate School in Ljubljana. His research interests are in machine learning, microarray data analysis, common sense knowledge representation and reasoning.

Filip Železný is the head of the Intelligent Data Analysis research group at the Gerstner Laboratory, Czech Technical University in Prague. He received his PhD in artificial intelligence and biocybernetics from the Czech Technical University and carried out post-doctoral training at the University of Wisconsin in Madison. He was a visiting professor at the State University of New York in Binghamton. Currently he is a grantee of the Czech Academy of Sciences and the European Commission. His main research interest is relational machine learning and its applications in bioinformatics.

Nada Lavrač is the head of the Department of Knowledge Technologies at the Jožef Stefan Institute, Ljubljana, Slovenia. She was the scientific coordinator of the European Scientific Network in Inductive Logic Programming (ILPNET, 1993-1996) and co-coordinator of the 5FP EU project Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise (SolEuNet, 2000-2003). She is an author and editor of numerous books and conference proceedings, including "Inductive Logic Programming: Techniques and Applications" (Kluwer 1994) and "Relational Data Mining" (Springer 2002). Her main research interests are in machine learning, relational data mining, knowledge management, and applications of intelligent data analysis in virtual organizations, medicine and bioinformatics.

Jakub Tolar received his M.D. from Charles University in Prague, Czech Republic, and his Ph.D. in Molecular, Cellular, Developmental Biology and Genetics from the University of Minnesota. He has been interested in the use of hematopoietic transplantation for bone marrow failure (e.g., aplastic anemia and dyskeratosis congenita) and metabolic disorders (e.g., mucopolysaccharidosis type I and adrenoleukodystrophy). His research focuses on the use of bone marrow derived stem cells and Sleeping Beauty transposon gene therapy for correction of genetic diseases and improving outcome of blood and marrow transplantation.