SUMMARIZING GENE-EXPRESSION-BASED CLASSIFIERS BY META-MINING COMPREHENSIBLE RELATIONAL PATTERNS

Filip Železný, Olga Štěpánková
Deptartment of Cybernetics
Czech Technical Univ. in Prague
Technická 2
166 27 Praha 6, Czech Republic
{zelezny,step}@labe.felk.cvut.cz

Jakub Tolar
Department of Pediatrics
Univ. of Minnesota Medical School
420 Delaware Street
55455 Minneapolis, USA
tolar003@umn.edu

Nada Lavrač
Department of Knowledge Technologies
Jožef Stefan Institute
Jamova 39
1000 Ljubljana, Slovenia
nada.lavrac@ijs.si

ABSTRACT
We propose a methodology for predictive classification from gene expression data, able to combine the robustness of high-dimensional statistical classification methods with the comprehensibility and interpretability of simple logic-based models. We first construct a robust classifier combining contributions of a large number of gene expression values, and then (meta)-mine the classifier for compact summarizations of subgroups among genes associated with a given class therein. The subgroups are described by means of relational logic features extracted from publicly available gene ontology information. The curse of dimensionality pertaining to the gene expression based classification problem due to the large number of attributes (genes) is turned into an advantage in the secondary, meta-mining task as here the original attributes become learning examples. We cross-validate the proposed method on two classification problems: (i) acute lymphoblastic leukemia (ALL) vs. acute myeloid leukemia (AML), (ii) seven subclasses of ALL.

KEY WORDS
Relational Data Mining, Gene Expression Microarrays, Gene Ontology

1 Introduction

Many tasks of automated knowledge discovery from gene expression microarray data by data mining algorithms aim at constructing classifiers able to diagnose a cancer type from a gene expression profile. See eg. the seminal papers [8, 17] for reference.

This task is characterized by the abundance of attributes (eg. simultaneously measured gene expression values) on one hand confronted with the shortage of the available samples (eg. patients/tissues subject to measurements) on the other hand. It is known from scientific discovery literature that such domains are prone to overfitting: overfitted classifiers are characterized by significantly decreased predictive accuracy on unseen samples compared to the training set accuracy.

Informally, in domains characterized by a small number of examples and a large number of attributes, overfitting occurs because some artefacts ("flukes") of actually irrelevant attribute combinations can emerge simply by means of chance and appear significant with respect to the examples available to the machine learning algorithm.

To avoid the overfitting pitfall, state-of-art approaches construct complex classifiers that combine relatively weak contributions of up to thousands of genes (attributes) to classify a disease. Real-valued support vector machine (SVM) [9] models are currently specifically popular in the gene expression data mining domain. However, classifiers based on many real-valued attributes have an important drawback: they are not appropriate for expert interpretation. The complexity of such classifiers limits their transparency and consequently the biological insight they can provide.

In [7], we have tested the feasibility of constructing simple yet robust logic-based classifiers, in the form of propositional rules amenable to direct expert interpretation, by an innovative algorithmic methodology of subgroup discovery. Such rules typically include two to five gene expression attributes and, in contrast to markers obtained from SVM schemes, these rules explicitly stress the importance of the correlation of the activity (or non-activity) of a narrow gene set. Followingly, further papers appeared [19, 15] proposing methods for constructing very simple, interpretable models for gene expression data.

Despite the obvious attractiveness stemming from the interpretability of the classifiers yielded by the above mentioned approaches, they still fall short of the high dimensional (attribute-rich) classifiers in terms of predictive accuracy in some benchmark subtasks considered in [7, 17]. Preliminary experiments [6] suggest that certain accuracy gap manifests itself also in the predictive discovery tasks formulated in [18].

We thus seem to face an inevitable trade-off between interpretability on one hand and accuracy on the other hand pertaining to predictive gene expression data based models. Here we propose a methodology with the potential to overcome this challenge, combining the advantages of both the approaches of high-dimensional models and the simple logic models.

The fundamental idea is as follows. First, a high-dimensional classification model C is constructed from gene expression data D, wherein each sample is assigned a class label c out of a set of class labels $\mathcal{C}$. Depending on the particular inductive method employed, C may acquire different forms, but as a general rule it will associate a high number of genes ('predictors') to each of the target classes in D; we denote this gene set by $G_C(c)$. In this paper we will confine ourselves to the straightforward way originally employed by [8], where $G_C(c)$ is simply a set of genes with expression highly correlated (positively or negatively) with the class c. However, generalizations where $G_C(c)$ would coincide with eg. the support vectors in a SVM classifier, principle components obtained through PCA, etc., are obvious.

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

The *subgroup discovery* procedure just outlined is approached as another supervised machine learning task. We refer to it as the secondary, or *meta-mining* task as it aims to mine from the outputs of the primary data mining process where genes of predictive strength are seeked. This secondary task is, in a way, orthogonal to the primary discovery process in that the original attributes (genes) now become learning examples, each of which has a class label $c \in \mathcal{C}$. To apply a discovery algorithm, information about relevant features of the new examples is required. No such features (ie. '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 (in this paper, we use the Entrez Gene database maintained at the US National Center for Biotechnology Information).

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. For example, a gene may be related to a variable number of cell processes, play role in variable numbers of regulatory pathways etc. This imposes 1-to-many relations hard to elegantly capture within an attribute set of fixed size. Furthermore, a useful piece information about a gene g may for instance be expressed by the feature

> 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. The difficulties of representing such features through attribute-value tuples is evident.

In summary, we are approaching the task of subgroup discovery from a relational data domain. For this purpose we employ the methodology of relational subgroup discovery we proposed in [14, 13] and implemented in the RSD algorithm [20]. Using RSD, we were able to discover knowledge such as

> *The expression of genes coding for proteins located in the integral to membrane cell component, whose functions include receptor activity, has a high correlation with the BCR class of acute lymphoblastic leukemia (ALL) and a low correlation with the other classes of ALL.*

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.[1] 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 [4].

2 Relational Feature Construction

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

```
interaction(g,G),
function(G,protein_binding)
```

Here, upper cases denote existentially quantified variables and g is the key term that binds a feature to a specific example (here a gene).

The user specifies a grammar declaration which constraints the resulting set of constructed features. RSD accepts feature language declarations similar to those used in the inductive logic programming system Progol [16]. A declaration lists the predicates that can appear in a feature, and to each argument of a predicate a *type* and a *mode* are assigned. In a correct feature, if two arguments have different types, they may not hold the same variable. A mode is either *input* or *output*; every variable in an input argument of a literal must appear in an output argument of some preceding literal in the same feature. [5] further dictate the opposite constraint: 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 optional constraints such as the maximum *variable depth* [16], maximum number of occurrences of a given predicate symbol in a feature, etc.

[1]This process is referred to as *propositionalization* [11]

RSD generates an exhaustive set of features satisfying the language declarations as well as a the *connectivity* requirement, which stipulates that 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(g,G1),
function(G1,protein_binding),
      interaction(g,G2),
   component(G2,membrane)
```

since it can be decomposed into two separate features. We do not construct such decomposable expressions, as these are redundant for the purpose of subsequent search for rules with conjunctive antecedents. Furthermore the concept of undecomposability allows for powerful search space pruning [14, 13]. Notice also that the expression above may be extended into an admissible undecomposable feature if a further logic atom is added:

```
      interaction(g,G1),
function(G1,protein_binding),
      interaction(g,G2),
   component(G2,membrane),
      interaction(G1,G2)
```

The construction of features is implemented as depth-first, general-to-specific search where refinement corresponds to adding a literal to the currently examined expression. During the search, each search node found to be a correct feature is listed in the output.

A remark is in turn concerning the way constants (such as `protein_binding`) are employed in features. Rather than making the user responsible for declaring all possible constants that may occur in features, RSD extracts them automatically from the learning data. The user marks the types of variables which should be replaced by constants. For each constant-free feature, a number of different features are then generated, each corresponding to a possible replacement of the combination of the indicated variables with constants. RSD then only proceeds with those combinations of constants, which make the feature true for at least a pre-specified number of examples.

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 Prolog language engine.

3 Subgroup Discovery

A subgroup discovery task is defined as follows: *Given a population of individuals and a property of individuals we are interested in, find population subgroups that are statistically 'most interesting', e.g., are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest.*

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 models (sets of rules), inducing class descriptions in terms of properties occurring in the descriptions of training examples, in contrast, subgroup discovery aims at discovering individual patterns of interest (individual rules describing the target class).

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

3.1 Inducing a single subgroup rule

Our algorithm is based on an adaptation of the standard propositional rule learner CN2 [3, 4]. 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[2] of an induced rule of the form $H \leftarrow B$ (where H is the rule head - 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: $p(H|B)$.

The accuracy heuristic $Acc(H \leftarrow B) = p(H|B)$ can be replaced by the *weighted relative accuracy* heuristic. Weighted relative accuracy is a reformulation of one of the heuristics used in MIDOS [21] aimed at balancing the size of a group with its distributional unusualness [10].

The weighted relative accuracy heuristic is defined as follows:

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

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

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

are estimated by relative frequencies[3] 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, $n(B)$ is the number of examples covered by rule $H \leftarrow B$, $n(H)$ is the number of examples of class H, and $n(HB)$ is the number of examples of class H correctly classified by the rule (true positives).

3.2 Inducing a set of subgroup 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 (ie. *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 interesting properties of subgroups of the entire population.

In contrast, RSD uses the *weighted covering algorithm*, which allows for discovering interesting subgroup properties in the entire 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 which is 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.

Initial weights of all positive examples e_j equal 1. In the first iteration of the weighted covering algorithm all target class examples have the same weight, while in the following iterations the contributions of examples are inverse proportional to their coverage by previously constructed rules; weights of covered positive examples thus 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.[4]

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

4 Experiments

This section presents a statistical validation of the proposed methodology. We do not access here the accuracy of disease classification from gene-expression values itself, as this is a property of the particular method used for the primary mining task, which is not our main concern. We rather aim at evaluating the properties of the secondary mining task. Namely, 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*, ie. a risk of discovering fluke subgroups. For this sake we compared the *precision* and *recall* values of the discovered subgroups on training sets with those on independent testing sets, in the frame of a 10-fold stratified cross-validation procedure.

4.1 Materials and methods

We apply the proposed methodology on two problems of predictive classification from gene expression data.

The first was introduced in [8] and aims at distinguishing between samples of acute lymphoblastic leukemia and acute myeloid leukemia from gene expression profiles obtained by the Affymetrix HU6800 microarray chip. The data contains 73 class-labelled samples of expression vectors.

The second was defined in [18]. Here one tries to distinguish among seven classes of pediatric acute lymphoblastic leukemia from gene expression profiles obtained by the Affymetrix HG-U133A microarray chip. The data contains 132 class-labelled samples.

In both data sets, for each class c we first extracted a set of genes $Pr(c)$ ($Ab(c)$) whose present (absent, respectively) expression is highly correlated with c. More precisely, for each gene g and class c we evaluated the functions f_T (for $T \in \{Pr, Ab\}$):

[3] Alternatively, the Laplace [2] and the m-estimate [1] could also be used.

[4] Whereas this approach is referred to as *additive* in [12], another option is the *multiplicative* approach, where for a given parameter $\gamma < 1$, weights of positive examples covered by i rules decrease according to γ^i. Both approaches have been implemented in RSD, but additive weights lead to better results.

Task	**Class**	**No. genes in $Pr(c)$**	**No. genes in $Ab(c)$**
AML/ALL	AML	40	73
	ALL	73	42
ALL subtypes	BCR	47	3
	E2A	80	318
	HD50	118	36
	MLL	130	109
	T-ALL	237	110
	TEL	187	95
	Other	66	54

Table 1. Numbers of genes in $Pr(c)$ ($Ab(c)$, respectively) extracted for each task and class c and used subsequently as examples in the meta-mining procedure.

$$f_T(g,c) = P_T(g)\left(\frac{P_T(c,g)}{P_T(g)} - P(c)\right) \tag{5}$$

Here, $P(c)$ denotes the probability that a randomly drawn sample falls into class c. $P_{Pr}(g)$ ($P_{Ab}(g)$) is the probability that g has present (absent) expression in a randomly drawn sample. $P_T(c,g)$ ($T \in \{Pr, Ab\}$) is the joint probability of both events. All probabilities are estimated as relative frequencies.

Note that the above function in fact implements the $WRAcc$ heuristic defined in Eq. 1. As such it estimates the predictive power of the presence (absence) of the expression of a gene g with respect to a target class c. We intentionally keep distinct notations in Eq's. 5 and 1 to avoid confusion between the $WRAcc$ application in the primary (gene selection) and the secondary (subgroup meta-mining) tasks.

We set a fixed threshold on $f_T(g,c)$ so that $\forall g, c$: $g \in Pr(c)$ ($g \in Ab(c)$) whenever $f_T(g,c) \geq 0.1$ (for the ALL-AML task, $T \in \{Pr, Ab\}$) and $f_T(g,c) \geq 0.05$ (for the ALL-subtypes task, $T \in \{Pr, Ab\}$) task. By choosing the thresholds we control the scale of the meta-mining task: with the elected values, a few tens to a few hundreds of genes are assigned to each class in both classification problems. The exact numbers of genes extracted for each class/problem are listed in Table 1.

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 combining the *biobase*, *annotate*, *hu6800* (for the AML-ALL task), and *hgu133a* (for the ALL-subtypes task) packages for the R system for statistical computing. The four packages are available from `http://www.bioconductor.org/` and R is available from `http://www.r-project.org/`.

Knowing the gene identifiers, the annotations can be accessed through hypertext queries to the Entrez Gene database, which is available at `http://www.ncbi.nlm.nih.gov/`. We developed a program script in the Python language, which automatically queries the server for the gene annotations, parses them and produces their structured, relational logic representations digestible by RSD. This script is available on request to the first author.

gene set	**Train** precision (st.dev.)	recall (st.dev.)	**Test** precision (st.dev.)	recall (st.dev.)
$Pr(c)$	0.96 (0.01)	0.28 (0.01)	0.89 (0.05)	0.23 (0.09)
$Ab(c)$	0.96 (0.01)	0.29 (0.02)	0.88 (0.06)	0.24 (0.09)

Table 2. Precision-recall figures for the AML-ALL classification task obtain through 10-fold stratified cross-validation.

gene set	**Train** precision (st.dev.)	recall (st.dev.)	**Test** precision (st.dev.)	recall (st.dev.)
$Pr(c)$	0.83 (0.02)	0.87 (0.04)	0.45 (0.06)	0.47 (0.05)
$Ab(c)$	0.85 (0.02)	0.87 (0.02)	0.49 (0.04)	0.40 (0.04)

Table 3. Precision-recall figures for the ALL-subtypes classification task obtain through 10-fold stratified cross-validation.

4.2 Results

We subjected the RSD algorithm to a 10-fold stratified cross-validation on both classification tasks. Within each fold, the first 10 subgroups produced by RSD from the training split were considered and their *precision* and *recall* values were computed on both the training split and the testing split. For both splits, the ten precision values as well as the ten recall values thus obtained were averaged. For one fold, we thus obtained four average values. Each of them was further averaged among all 10 folds following the standard cross-validation regime.

Tables 2 and 3 show the final average (along with standard deviation figures) results for the two respective classification tasks, and for both versions of the primary gene selection procedures (based on present/absent expression).

We observe that no significant overfitting effect manifests itself in the AML-ALL task; the training and testing figures are close and rather favorable. Although the results for the ALL-subtypes classification are quite satisfactory in absolute values (given there are 7 classes, the baseline random choice precision would be about 14%), the gap between the performances on training data on one hand and testing data on the other hand is significant. A possible reason may be that the threshold used for gene inclusion into the $Pr\,(Ab)$ sets was too high, resulting in example sets not large enough to allow for induction of stable relational descriptions in this domain. Repeating the experiments with a lower threshold will thus be the first step in our future work.

5 Discussion

In this paper we have proposed a methodology for predictive classification from gene expression data, able to combine the robustness of high-dimensional statistical classification methods with the comprehensibility and interpretability of simple logic-based models. Our methodology proposes to first construct a robust classifier combining contributions of a large number of gene expression values, and then finding compact, relational descriptions of subgroups among genes employed in the classifier.

It is noteworthy that the 'post-processing' step is also a machine learning task, in which the curse of dimensionality (the number of attributes – gene expressions measured) usually ascribed to the type of classification problem considered, actually turns into an advantage. The high number of *attributes*, incurring the risk of overfitting, turns into a high number of *examples*, which on the contrary works *against* overfitting in the subsequent subgroup discovery task. 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.

The statistical evaluation conducted within the present study is not entirely conclusive (basic trends differ among the two domains tested) as to the potential inclination of the methodology to discover fluke subgroups and more experiments are needed for a reliable assessment. It has to be noted though that the subgroup descriptions provided by RSD are typically intuitively convincing from the biological point of view even in the ALL-subtype classification where the statistical evaluation results are less favorable. For example, the following subgroup description was produced for the BCR class (we convert the relational logic representation to natural language for ease of reading):

> BCR class: *Genes coding for proteins located in the integral to membrane cell component, whose functions include receptor activity.*

Indeed, BCR/abl is a classic example of a leukemia driven by spurious expression of a fusion protein expressed as a continuously active kinase protein on the membrane of leukemic cells. The second subgroup example is for the TEL class in the same classification problem:

> TEL class: *Genes coding for proteins located in the nucleus whose functions include protein binding and whose related processes include transcription.*

By contrast to BCR, the TEL leukemia is driven by expression of a protein, which is a transcription factor active in the nucleus. As a result, these findings related to the location, function and processes associated to the subgroups, represent the most salient features of these respective types of acute lymphoblastic leukemia.

In summary, we have high hopes on discovering novel, yet reliable knowledge from the relational combination of gene expression data with public gene annotation databases in future applications of our methodology.

6 Acknowledgements

F.Ž. is grateful to the Grant Agency of the Czech Academy of Sciences (CAS) for the support through the project KJB201210501 Logic Based Machine Learning for Genomic Data Analysis. J. T. is supported by Children Cancer Research Fund and University of Minnesota Cancer Center. O.Š. is supported by CAS through the project 1ET101210513 Relational Machine Learning for Biomedical Data Analysis.

References

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

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

[3] P. Clark and T. Niblett. Induction in noisy domains. In *Progress in Machine Learning (Proceedings of the 2nd European Working Session on Learning)*, pages 11–30. Sigma Press, 1987.

[4] P. Clark and T. Niblett. The cn2 induction algorithm. *Machine Learning*, pages 261–283, 1989.

[5] P. Flach and N. Lachiche. 1BC: A first-order Bayesian classifier. In S. Džeroski and P. Flach, editors, *Proceedings of the 9th International Workshop on Inductive Logic Programming*, pages 92–103. Springer, 1999.

[6] D. Gamberger. (personal communication).

[7] D. Gamberger, N. Lavrač, F. Železný, and J. Tolar. Induction of comprehensible models for gene expression datasets by subgroup discovery methodology. *Jr Biomedical Informatics*, 37(5):269–284, 2004.

[8] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer:

Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[9] T. Hastie and J. Tibshirani, R. andd Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[10] W. Kloesgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in Knowledge Discovery and Data Mining*, pages 249–271. AAAI Press, Menlo Park, CA, 1996.

[11] N. Lavrač and P. A. Flach. An extended transformation approach to inductive logic programming. *ACM Transactions on Computational Logic*, 2(4):458–494, October 2001.

[12] N. Lavrač, B. Kavšek, P. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.

[13] N. Lavrač and F. Železný. Propositionalization-based relational subgroup discovery with RSD. *Machine Learning*. (to appear in the 2005 special issue on statistical relational learning, paper available at http://labe.felk.cvut.cz/∼zelezny/pubs/mlj05.pdf).

[14] 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*, pages 149–165. Springer, 2002.

[15] M. Mramor, G. Leban, J. Demsar, and B. Zupan. Conquering the curse of dimensionality in gene expression cancer diagnosis: Tough problem, simple models. In *Proceedings of the 10th Conference on Artificial Intelligence in Medicine*, pages 514–523, 2005.

[16] S. Muggleton. Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4):245–286, 1995.

[17] 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 Science USA*, volume 98, pages 15149–54, 2001.

[18] 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 profiling. *Blood*, 102(8):2961–9, 2003.

[19] S. A. Vinterbo and L. Kim, E. Y. Ohno-Machado. Small, fuzzy and interpretable gene expression based classifiers. *Bioinformatics*, 21(9):531–537, 2005.

[20] F. Železný. RSD user's manual. Available at: http://labe.felk.cvut.cz/∼zelezny/rsd/rsd.pdf.

[21] S. Wrobel. An algorithm for multi-relational discovery of subgroups. In Jan Komorowski and Jan Zytkow, editors, *Proceedings of the First European Symposion on Principles of Data Mining and Knowledge Discovery (PKDD-97)*, pages 78–87. Springer, 1997.