Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Biomedical Informatics 37 (2004) 269–284
Journal of Biomedical Informatics
www.elsevier.com/locate/yjbin

Induction of comprehensible models for gene expression datasets by subgroup discovery methodology

Dragan Gamberger[a,*], Nada Lavrač[b,c], Filip Železný[d,e], Jakub Tolar[f]

[a] Laboratory for Information Systems, Rudjer Bošković Institute, Bijenička 54, 10000 Zagreb, Croatia
[b] Department of Knowledge Technologies, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[c] Nova Gorica Polytechnic Vipavska 13, 5000 Nova Gorica, Slovenia
[d] Department of Cybernetics, Czech Institute of Technology (CVUT FEL), Technická 2, 16627 Prague, Czech Republic
[e] Department of Biostatistics, University of Wisconsin Medical School, 1300 University Avenue, 53706 Madison, USA
[f] Institute of Human Genetics, University of Minnesota Medical School, 420 Delaware Street, 55455 Minneapolis, USA

Received 14 June 2004

Abstract

Finding disease markers (classifiers) from gene expression data by machine learning algorithms is characterized by a high risk of overfitting the data due the abundance of attributes (simultaneously measured gene expression values) and shortage of available examples (observations). To avoid this pitfall and achieve predictor robustness, state-of-the-art approaches construct complex classifiers that combine relatively weak contributions of up to thousands of genes (attributes) to classify a disease. The complexity of such classifiers limits their transparency and consequently the biological insights they can provide. The goal of this study is to apply to this domain the methodology of constructing simple yet robust logic-based classifiers amenable to direct expert interpretation. On two well-known, publicly available gene expression classification problems, the paper shows the feasibility of this approach, employing a recently developed subgroup discovery methodology. Some of the discovered classifiers allow for novel biological interpretations.

Keywords: Gene expression measurements; Disease markers; Subgroup discovery; Machine learning; Comprehensible classification

1. Introduction

Gene expression monitoring by DNA microarrays (gene chips) provides an important source of information that can help in understanding many biological processes. This technology allows for novel applications, resulting in increased understanding of disease processes, and improved diagnosis and prediction in medicine.

Data collected in these applications are not suitable for direct human explanatory analysis because a single DNA microarray experiment results in thousands of measured expression values and also because of the lack of existing expert knowledge available for the analysis. The application of various data mining and knowledge discovery methods using machine learning algorithms [39] seems an evident approach to take in such a problem domain. Numerous approaches have been suggested towards exploiting state-of-the-art machine learning or microarray data mining, including both supervised learning (learning from data with class labels) and unsupervised learning (such as conceptual clustering). A state-of-the-art review of these various approaches can be found in [15,40].

* Corresponding author. Fax: +385 1 4680 114.
E-mail addresses: dragan.gamberger@irb.hr (D. Gamberger), nada.lavrac@ijs.si (N. Lavrač), zelezny@fel.cvut.cz, zelezny@biostat.wisc.edu (F. Železný), tolar003@umn.edu (J. Tolar).

1532-0464/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2004.07.007

In this study, we follow the supervised learning paradigm. The database we analyze consists of a set of gene expression measurements (examples), each corresponding to a rather large number of measured expression values of a predefined family of genes (attributes). Each measurement in the database was extracted from a tissue of a patient with a specific disease; this disease is the class for the given example. The standard goal of machine learning is to start from such available labeled examples and construct classifiers that can successfully classify new, previously unseen examples. Such classifiers are important because they can be used for diagnostic purposes in medicine and because they can help to understand the dependencies between classes (diseases) and attributes (gene expression values).

The problem of finding disease markers (classifiers) from gene expression data by machine learning algorithms is characterized by the abundance of attributes (simultaneously measured gene expression values), and the shortage of the available examples (patients subject to measurements). The application of machine learning algorithms in a domain characterized by a large number of attributes typically calls for some dimensionality reduction, even if the employed strategy can in principle directly accept all the available attribute values. The benefits of prior elimination of irrelevant (or weakly relevant) attributes in data preprocessing has been recognized in machine learning [35]: besides helping to reduce the problem complexity and the computation time, it can enable the construction of more accurate classifiers.

From the dimensionality point of view, the gene expression domain is specifically unfavorable, because—as we have mentioned—the abundance of attributes is confronted with a relatively small number of available examples. It is known from the machine learning and 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, or—in other words—by a high generalization error [14]. See [21] for an extensive treatment of different effects of overfitting. Informally, in domains characterized by a small number of examples and a large number of attributes, overfitting occurs because some artifacts (flukes) of actually irrelevant attribute combinations can emerge simply by means of chance and appear significant with respect to the examples available to a machine learning algorithm.

To avoid the overfitting pitfall, state-of-the-art approaches construct complex classifiers that combine relatively weak contributions of up to thousands of genes (attributes) to classify a disease [20,45,10,36]. Predictor robustness is achieved by the redundancy of classifiers, realized, e.g., by voting of multiple classifiers. For example, [20] use weighted voting of informative genes, [45,10] employ the support vector machine (SVM) paradigm, while in [36] scores of top ranked emerging patterns are used. The achieved prediction quality on independent test sets are very high but a drawback of classifiers based on many attributes is that they are not appropriate for expert interpretation. Although it is possible to extract the attributes with maximal voting weight (in [45] such genes are called *disease markers* and some of them are already identified as useful in routine clinical practice), the logical connections among the extracted attributes are lost and the construction of expert comprehensible (disease) models remains a very difficult task.

This paper describes an approach to the detection of rules for the classes of gene expression samples that are much more convenient for expert interpretation, taking gene expression data modeling as a novel challenge for the application of the recently developed subgroup discovery methodology [19]. Its goal is the induction of classifiers in the form of explicit short rules describing important subgroups of the target class samples, although these simple classifiers may be of a lower predictive quality than the more complex classifiers. Induced rules typically include 2–5 gene expression attributes and, in contrast to markers obtained from voting schemes, these rules explicitly stress the importance of the correlation of the activity (or non-activity) of genes in the selected set of attributes. The problem with the induction of low dimensional, non-redundant classifiers is that they are prone to overfitting the training set. The selection of an appropriate hypothesis language as well as the reduction of the hypothesis search space are known methods for avoiding overfitting [46]. Handling overfitting by relevancy based feature and rule filtering are important aspects of this work. In rule learning, the problem of this approach is that with a strongly reduced hypothesis space it may be difficult to induce rules that cover all/many examples from the training set. However, the proposed subgroup discovery approach provides a much better framework for the application of the suggested methodology of feature and rule relevancy than the standard separate-and-conquer rule learning [18].

To arrive at simple rule-based predictors, the numeric microarray data are discretized, i.e., represented by means of categorical values. This admittedly introduces a superfluous degree of freedom in the choice of the discretization threshold values. We adhere to what is apparently the most natural choice—the discretization provided by Affymetrix, the microarray manufacturer—and it is part of our study to test whether interesting knowledge can be discovered with such discretized data. Naturally, by selecting this approach the risk of overfitting, potentially leading to rules that do not reflect genuine dependencies between classes and gene activity values, is not automatically avoided.

The paper outline is as follows. In Section 2, the subgroup discovery approach is presented, together with

techniques aimed at avoiding overfitting. Section 3 discusses the results obtained on two publicly available gene expression problem domains. Expert interpretation of a subset of rules with high predictive value confirmed on independent test sets is provided in Section 4, showing the significance of the discovered relationships.

2. The subgroup discovery methodology

This section presents the subgroup discovery methodology[1] originally introduced in [19]. Subgroup discovery is a form of supervised inductive learning of subgroup descriptions of a given target class. The descriptions have the form of rules built as logical combination of features. Features are logical conditions that have values *true* or *false*, depending on the values of attributes which describe the examples of the given problem domain. Subgroup discovery rule learning is therefore a type of two-class attribute based (zero order) inductive learning. Multi-class problems can be solved as a series of two-class problems, so that in each run one class is selected as the target class while examples of all other classes are treated as non-target class cases.

There is a large body of previous research on rule induction in machine learning and data mining, and on subgroup discovery in general. We refer the reader to the mentioned source [19] explaining how the SD algorithm relates to similar algorithms, such as classification [11,38] and association [2,25] rule learners and subgroup discovery systems [31,54].

For the purpose of the induction of subgroup descriptions for gene expression datasets, the system has been enhanced to be able to accept datasets with a larger number of attributes. For performing applications in gene expression data analysis, additional techniques for handling overfitting have been implemented, described in detail in this section.

2.1. An outline of the subgroup discovery approach

Subgroup discovery [54,19] has the goal to uncover characteristic properties of population subgroups by building short rules which are highly significant (assuring that the distribution of classes of covered instances are statistically significantly different from the distribution in the training set) and have a large coverage (covering many target class instances).

In this work, subgroup discovery is performed by the SD algorithm, a relatively simple iterative beam search rule learning algorithm [19]. The SD input consists of a set of examples E ($E = P \cup N$, P is the set of target class examples and N the set of non-target class examples) and the set of features F that are constructed for the given example set. For discrete (categorical) attributes, features have the form $Attribute = value$ or $Attribute \neq value$, while for continuous (numerical) attributes they have the form $Attribute > value$ or $Attribute \leqslant value$. The output of the SD algorithm is a set of rules with optimal covering properties on the given example set. As in classification rule learning, an induced rule (subgroup description) has the form of a (backwards) implication: $Class \leftarrow Cond$. In terms of rule learning, the property of interest for subgroup discovery is the target class (*Class*) that appears in the rule consequent, and the rule antecedent (*Cond.*) is a conjunction of features (attribute–value pairs) selected from the features describing the training instances.

In the SD algorithm, subgroups are described by rules formed of conjunctions of a small number of features. Each rule describing a subgroup is extended with the information about the rule *quality* which enables the evaluation of induced rules. The output rule form is as follows:

$$Class \leftarrow Cond[Sens, Spec],$$

where *Class* is the target property of interest, *Cond.* is a conjunction of features, *Sens* is the *sensitivity* or *true positive rate*, i.e., the fraction of positive cases that are correctly classified as positive, computed as $|TP|/|P|$, and *Spec* is the *specificity* or *true negative rate*, i.e., the fraction of negative cases correctly classified as negative, computed as $|TN|/|N|$, for TP and TN being the sets of true positives (target class examples covered by a rule) and true negatives (non-target class examples not covered by the rule), respectively. Non-target class examples covered by the rule are called *false positives*, FP, and $N = TN \cup FP$.

Features, formed of attribute–value pairs, are constructed in the preprocessing step of the SD algorithm. To formalize the feature construction procedure, let values v_{ix} ($x = 1,\ldots,k_{ip}$) denote the k_{ip} different values of attribute A_i that appear in the target class examples and w_{iy} ($y = 1,\ldots,k_{in}$) the k_{in} different values of A_i appearing in the non-target class examples. A set of features F is constructed as follows:

- For discrete attributes A_i, features of the form $A_i = v_{ix}$ and $A_i \neq w_{iy}$ are generated.
- For continuous attributes A_i features of the form $A_i \leqslant (v_{ix} + w_{iy})/2$ are created for all neighboring value pairs (v_{ix}, w_{iy}), and features $A_i > (v_{ix} + w_{iy})/2$ for all neighbor pairs (w_{iy}, v_{ix}).

2.2. Handling overfitting

There is no ideal solution to the problem of data overfitting. No inductive learning algorithm can

[1] The approach has been implemented in the on-line Data Mining Server (DMS), publicly available at http://dms.irb.hr. DMS and its constituting subgroup discovery algorithm SD can be tested on user submitted domains with up to 250 examples and 50 attributes.

guarantee that the induced rules will not overfit the training set. There are two main mechanisms that can be used to avoid overfitting.

- Overfitting can be reduced if the hypothesis search space is suitably restricted [14].
- In rule learning, the standard approaches to handling the problem of overfitting is through the use of appropriate search heuristics and stopping criteria used in rule construction, stopping criteria used in ruleset construction and rule truncation. For example, most separate-and-conquer based rule learners [18] (e.g., AQ, CN2, RIPPER, and CLASS) use heuristics aimed at maximizing rule accuracy. To avoid overfitting, these systems are capable of learning 'impure' rules with increased rule coverage (generality).

Accordingly, we implement both of these mechanisms in the employed subgroup discovery methodology.

- The hypothesis search space is restricted in three ways: through domain specific restrictions for feature construction for functional genomics domains, outlined in Section 2.3, filtering of irrelevant features described in Section 2.4 and filtering of irrelevant rules described in Section 2.5. In the implementation of these mechanisms, cautiousness is needed as strong restrictions of the hypothesis search space may prevent finding all the important rules. An equally important part of the methodology for avoiding overfitting is that each feature that enters the subgroup discovery process should itself be a relevant target class descriptor.
- Increased rule coverage, resulting in rules covering also non-target class examples, is achieved in the SD subgroup discovery algorithm by using the following rule quality measure in heuristic search: $|TP|/|(FP| + g)$, where g is a user defined *generalization parameter*. High quality rules will cover many target class examples and a low number of non-target examples. The number of tolerated negative examples, relative to the number of covered target class cases, is determined by parameter g. The SD beam search rule learning algorithm is described in Section 2.6.

2.3. Domain specific feature construction

Gene expression scanners measure signal intensity as continuous values which form an appropriate input for data analysis. The problem is that for continuous valued attributes there can be potentially many boundary values separating the classes, resulting in many different features for a single attribute. There is also a possibility to use presence call (signal specificity) values computed from measured signal intensity values by the Affymetrix GENECHIP software. The presence call has discrete values A (absent), P (present), and M (marginal). The M value can be interpreted as a '*do not know state*' and for the remaining values A and P it holds that feature $Attribute = A$ is identical to $Attribute \neq P$; consequently, for every attribute there are only two distinct features $Attribute = A$ and $Attribute = P$ generated for each gene.[2]

Signal intensity values are most frequently used [36] because they impose less restrictions to the classifier construction process and because the results do not depend on the GENECHIP software presence call computation. In the subgroup discovery approach, we prefer the use of presence call values. The reason is that features presented by conditions like gene A_i is present or gene A_j is absent are very natural for human interpretation. Although the GENECHIP software presence call computation may not be ideal, expert evaluation of the results demonstrates that it can enable induction of very interesting rules both because of the ease of their interpretation and because of their predictive quality.

A more important reason for using presence call values is that the approach can help in avoiding overfitting, as the feature space is very strongly restricted: instead of many features per attribute we have only two. Also, as the measured gene expression values are not completely reliable (which is reflected by the fact that for the same sample measured values may change from one measurement to another), some robustness of constructed rules is welcome. To some extent, this can be achieved by treating the marginal presence call attribute value M as a '*do not know*' state. The value can neither be used to support the relevancy of a feature or a rule, nor can it be used for prediction purposes. In this way, it additionally restricts the hypothesis search space.

A drawback of this approach is that we depend on the GENECHIP software presence call computation which can change with time. However, the SD methodology is general in the sense that it can accept as its input either $A/P/M$ values computed by any software, or real signal intensity values.

With respect to the feature construction process, the following observations are worth reflecting on. The features are restricted to simple forms only, as defined in Section 2.1, because their complex forms may enable that, despite testing feature covering properties, features with insufficient supportive evidence may enter the rule construction process. For example, for discrete attributes the simple features have the form $A_i = a$ or $A_i \neq a$. No complex logical forms like $(A_i = a \wedge A_j = b)$ or $(A_i = a \vee A_j = b)$ are acceptable. The first form is not needed as all potential conjunctions are tested by

[2] See Table 1 to see how the '*do not know*' value M is handled in the preprocessing of the SD algorithm.

the beam search procedure of the subgroup discovery algorithm. The second form is dangerous because, for example, the feature $A_i = a$ may be relevant while the feature $A_j = b$ may be irrelevant. Their combination $A_i = a \vee A_j = b$ may be even more relevant than $A_i = a$ itself, which may cause that condition $A_j = b$ may be included into the finally constructed rules while its inclusion is not justified by its covering properties on the training set. Notice that if both conditions $A_i = a$ and $A_j = b$ are relevant, it does not mean that by restricting the form of used features some important logical combinations of features will be ignored. In the subgroup discovery approach, both features can build separate subgroup descriptions and—if they are relevant—they both have a chance to appear in the final set of induced rules.

2.4. Feature filtering

Features are elementary ingredients of rules. But individual features are short rules themselves. The quality of a feature is determined by its covering property on the training set. This section presents the methodology enabling the detection and elimination of irrelevant features which significantly helps in reducing the hypothesis space. More importantly, the methodology ensures that only relevant features will enter the process of rule construction which is important for avoiding overfitting. Although this section mentions only feature filtering, the same methodology is applicable to any logical combination of features, including the complete rules.

Definition 1. (Total irrelevancy.) A feature that has either $|TP| = 0$ or $|TN| = 0$ is totally irrelevant.

If a feature has $|TP| = 0$ or $|TN| = 0$ it is totally irrelevant because it is of no use in building rules that distinguish one class from the other. A gene is called *constant-valued gene* if it has, besides some M values, either only A or only P values for all examples in the training set. Both features generated from a constant-valued attribute are totally irrelevant because they either have $|TP| = 0$ or $|TN| = 0$. Table 1 presents a constant-valued attribute and the features generated for the given attribute. In the experiments presented in Section 3, the number of detected constant valued attributes eliminated in preprocessing was between 12 and 40%.

In the example in Table 1, it can be also noticed that feature f has value *false* for the attribute value M when the example is in the target class and value *true* when the example is in the non-target class. It means that for an example with attribute value M, feature truth-values do not depend on the properties of the feature but on the class to which the example belongs.

While total irrelevancy helps in reducing the computational complexity of the machine learning task, the main goal of applying absolute feature irrelevancy is to ensure a minimal quality of features which are used in the rule induction process.

Definition 2. (Absolute irrelevancy.) A feature that has either $|TP| < min\_tp$ or $|TN| < min\_tn$ is absolutely irrelevant, where min_tp and min_tn are user defined constraints.

A feature with $|TP| < min\_tp$ is true for a small number of target class examples and a feature with $|TN| < min\_tn$ is false for a small number of non-target class examples. It is assumed that such small numbers may be due to statistical chance so that it seems reasonable not to use features with either of these properties in the rule construction process. Through a conjunctive connection of features, the generated rule will have $|TP|$ smaller or equal than the smallest $|TP|$ value of the features forming the subgroup description. In contrast, the rule $|TN|$ value will be at least as large as the largest $|TN|$ of the used features. This is the reason why min_tp is typically larger than min_tn and it can be as large as the minimal estimated number of samples that must be covered by any acceptably good subgroup for the domain.

The problem with absolute irrelevancy is that both min_tp and min_tn are user defined constants. Optimal values for these constants may significantly change from one application to another. A practical suggestion is to start with small values of these constants and experiment by increasing the values. Our experience in gene expression domains suggests to choose $min\_tp = |P|/2$ and $min\_tn = \sqrt{|N|}$ as the starting values; these values have been used in all the experiments reported

Table 1
A table illustrating five positive and four negative examples for a given target class (the selected cancer type) in which gene X is constant-valued and both features X = A and X = P are totally irrelevant

	Target class samples					Non-target class samples			
	M	*A*	*A*	*A*	*A*	*A*	*A*	*M*	*A*
Gene X									
Feature $X = A$	False	True	True	True	True	True	True	True	True
Feature $X = P$	False	False	False	False	False	False	False	True	False

The first has $|TN| = 0$ while the second has $|TP| = 0$.

in Section 3. In these experiments, the number of detected absolutely irrelevant features was between 50 and 75%.

While the aim of using absolute relevancy is to ensure a minimal quality that must be satisfied by every feature, relative relevancy should ensure that only the best among the available features will enter the rule construction process.

Definition 3. (Relative irrelevancy.) Feature f is irrelevant if there exists another feature f_{rel} such that true positives of f are a subset of true positives of f_{rel}, $TP(f) \subseteq TP(f_{rel})$, and true negatives of f are a subset of true negatives of f_{rel}, $TN(f) \subseteq TN(f_{rel})$.

If for feature f there exists another feature f_{rel} with the property that if in any rule f is substituted by f_{rel}, the rule quality measured by the number of correct classifications on the example set does not decrease, then it means that f_{rel} can be always used instead of f, and that we actually do not need f. Relative irrelevancy is very useful because it does not depend on user defined threshold values and its usage is suggested for all machine learning approaches [34].

If genes are described by A, P, and M values and if there are two genes with identical values for all training examples then one of them can be eliminated as irrelevant because the features based on this gene have the same covering properties as the features based on the other gene. If two genes X and Y do not have identical values for all examples then if it happens that one of the features of X is irrelevant because of the feature constructed as a condition of gene Y then the other feature of X may not be irrelevant because of the other feature of Y. But this property does not mean that attributes can be eliminated only if there exists another attribute with identical values; there can exist another gene Z whose feature will make the second feature of X irrelevant and make the complete attribute X irrelevant as well. This is demonstrated by an example in Table 2. It is also possible that the second feature is absolutely irrelevant because of a small $|TP|$ or $|TN|$ value.

In cases when continuous gene expression values are used, the same conditions for feature relevancy are applicable. But there are many features constructed from a single gene and all of them must be detected as irrelevant in order that a complete gene is eliminated because of its irrelevancy.

In the experiments presented in Section 3, the process of identifying relative irrelevancy eliminated between 65 and 85% of features. Most of them are detected also as absolutely irrelevant features. By the combination of the conditions for absolute and relative irrelevancy it was possible to eliminate 70–90% of features that remained after the elimination of totally irrelevant features. The importance of the approach is that the remaining features satisfy some predefined quality (determined by the absolute relevancy condition), and more importantly, that they are the best features for the domain (according to the relative relevancy criterion).

The reader may wonder why the preselection of features is done in a univariate way. At a first glance it may seem possible that two feature having the TP and TN properties of a coin toss when viewed in isolation exhibit strong predictive power in combination (as is the case in predicting x-or). It can easily be shown that if feature f is relatively irrelevant because of feature f_{rel} and feature g is relatively irrelevant because of feature g_{rel}, then $f \wedge g$ is relatively irrelevant because of $f_{rel} \wedge g_{rel}$. This claim can be verified by first fixing one of the two conjuncts, e.g., $g_{rel} = g$ and showing that in this case $TP(f \wedge g) \subseteq TP(f_{rel} \wedge g)$ and $TN(f \wedge g) \subseteq TN(f_{rel} \wedge g)$. Next, the same relationship can be shown also for the case when g is relatively irrelevant because of g_{rel}. Consequently, if for feature f there exists another feature f_{rel} with the property that if in any rule f is substituted by f_{rel} the rule quality measured by the number of correct classifications $|TP|$ and $|TN|$ does not decrease, then it means that f_{rel} can be always used instead of f, and that we actually do not need f. This means that f can be eliminated as irrelevant. Hence, the filtering of relatively irrelevant features will not hinder the construction of relevant conjuncts.

Table 2
An example in which gene X is relatively irrelevant because its feature $X = A$ is relatively irrelevant because of feature $Y = A$, and its feature $X = P$ is relatively irrelevant because of feature $Z = A$

	Target class samples					Non-target class samples			
	A	*P*	*A*	*P*	*M*	*A*	*P*	*M*	*P*
Gene X									
Feature $X = A$	True	False	True	False	False	True	False	True	False
Feature $X = P$	False	True	False	True	False	False	True	True	True
Gene Y	*A*	*P*	*A*	*P*	*A*	*P*	*P*	*M*	*P*
Feature $Y = A$	True	False	True	False	True	False	False	True	False
Feature $Y = P$	False	True	False	True	False	True	True	True	True
Gene Z	*A*	*A*	*P*	*A*	*A*	*P*	*P*	*A*	*A*
Feature $Z = A$	True	True	False	True	True	False	False	True	True
Feature $Z = P$	False	False	True	False	False	True	True	False	False

2.5. Rule filtering

Any rule induced by the SD algorithm must have at least a minimal support.[3] Minimal acceptable support for a domain is defined by the user defined *min_support* parameter (see the SD algorithm in Section 2.6). If a subrule of a rule (a subset of features forming the rule) does not satisfy this condition then the rule as a whole does not satisfy it either. Therefore, this condition is built into the iterative loop of the SD algorithm and every partial solution of best features which does not satisfy this condition cannot be kept in the beam. Besides restricting the search space, this requirement enables shorter algorithm execution time. The default value for the parameter *min_support* is equal to $\sqrt{|P|}/|E|$, but for the gene expression data it can be as high as $|P|/2|E|$. The *min_support* condition is tested in step 7 of the SD algorithm described in Section 2.6.

High confidence[4] of induced rules is ensured by the definition of rule quality q_g used in the search process of the SD algorithm (Section 2.6) which prefers rules with large $|TP|$ and small $|FP|$. Although the length of induced rules is not limited, the approach ensures the construction of short rules; the reason is that conjunctions of features have the property that the number of target class examples covered by adding a conjunct to a conjunction of features decreases. Long conjunctive rules have a very small chance to satisfy the minimal support condition and to be optimal with respect to rule quality q_g at the same time. In the experiments, all the induced rules have up to four features while all those explicitly shown in this paper and analyzed by the domain expert have only two features. This is very favorable because the complexity of the hypothesis space is significantly restricted and enables easy expert analysis.

2.6. Algorithm SD

The goal of the subgroup discovery algorithm SD, outlined in Fig. 1, is to search for rules that maximize rule quality measure $q_g = \frac{|TP|}{|FP|+g}$. High quality rules cover many target class examples and a low number of non-target examples. The user can express his preferences about rule generality (how many target class cases are covered by the rule description) in respect to the rule specificity (how many non-target class cases are covered by the rule) by selecting the parameter g. For low g values ($g \leqslant 1$), induced rules will have high specificity since every false positive classification is made relatively very 'expensive'. On the other hand, by selecting a high g value ($g > 10$ for small domains), more general rules will be generated which can have also many false positive predictions. Suggested g values in the SD algorithm in the Data Mining Server are in the range between 0.1 and 100, for analyzing data sets of up to 250 examples.

In addition to parameters g and *min_support*, the SD algorithm has an additional parameter which is defined by the user, but which does not need to be adjusted frequently. The *beam_width* parameter (default value is 100 for gene expression domains) defines the number of solutions kept in the beam in each iteration. The output of the algorithm is set S of *beam_width* different rules with highest q_g values. In the described experiments, we have used only the first (best) solution although there is a possibility to select a few relatively different solutions using the algorithm described in [19], or to enter the expert evaluation process with a set of a few best rules, letting the experts select the optimal solution(s). Moreover, the rules from set S could be used as an input to a redundant voting classifier, but this variant is out of the scope of this work.

The algorithm initializes all the rules in *Beam* and *New_beam* by empty rule conditions. Their quality values $q_g(i)$ are set to zero (step 1). Rule initialization is followed by an infinite loop (steps 2–12) that stops when, for all rules in the beam, it is no longer possible to further improve their quality. Rules can be improved by conjunctively adding features from F. After the first iteration, a rule condition consists of a single feature, after the second iteration up to two features, and so forth. The search is systematic in the sense that for all rules in the beam (step 3) all features from F (step 4) are tested in each iteration. For every new rule, constructed by conjunctively adding a feature to rule body (step 5), quality q_g is computed (step 6). If the support of the new rule is greater than *min_support* and if its quality q_g is greater than the quality of any rule in *New_beam*, the worst rule in *New_beam* is replaced by the new rule. The rules are reordered in *New_beam* according to their quality q_g. At the end of each iteration, *New_beam* is copied into *Beam* (step 11). When the algorithm terminates, the first rule in *Beam* is the rule with maximum q_g.

A necessary condition (in step 7) for a rule to be included in *New_beam* is its relative relevancy. A new rule is irrelevant if there already exists a rule R in *New_beam* such that true positives of the new rule are a subset of true positives of R and true negatives of the new rule are a subset of true negatives of R (in the same way as relative feature irrelevancy described in Section 2.4). After the new rule is included in *New_beam* it may happen that some of the existing rules in *New_beam* become relatively irrelevant with respect to this new rule. Such rules are eliminated from *New_beam* during its reordering (in step 8). The testing of relevancy ensures that *New_beam* contains only different and relatively relevant rules.

[3] Support is the number of correctly classified target class samples divided by the total number of samples, $|TP|/|E|$.

[4] Confidence (also called precision) is the fraction of all samples classified into the target class that actually belong to the target class, $|TP|/(|TP| + |FP|)$.

Algorithm SD: Subgroup Discovery
Input: $E = P \cup N$ (E training set, $|E|$ training set size,
P positive (target class) examples, N negative (non-target class) examples)
F set of all defined features, $f \in F$
Parameter: g (generalization parameter, $0.1 < g$, default value 1)
min_support (minimal support for rule acceptance)
beam_width (maximal number of rules in *Beam* and *New_Beam*)
Output: $S = \{TargetClass \leftarrow Cond\}$ (set of rules formed of *beam_width* best conditions *Cond*)
(1) **for** all rules in *Beam* and *New_Beam* ($i = 1$ to *beam_width*) **do**
initialize condition part of the rule to be empty, $Cond(i) \leftarrow \{\}$
initialize rule quality, $q_g(i) \leftarrow 0$
(2) **while** there are improvements in *Beam* **do**
(3) **for** all rules in *Beam* ($i = 1$ to *beam_width*) **do**
(4) **for** all $f \in F$ **do**
(5) form a new rule by forming a new condition as a conjunction of the condition from *Beam* and feature f, $Cond(i) \leftarrow Cond(i) \wedge f$
(6) compute the quality of a new rule as $q_g = \frac{|TP|}{|FP|+g}$
(7) **if** $\frac{|TP|}{|E|} \geq min\_support$ **and if** q_g is larger than any $q_g(i)$ in *New_Beam* **and if** the new rule is relatively relevant **do**
(8) replace the worst rule in *New_Beam* with the new rule and reorder the rules in *New_Beam* with respect to their quality
(9) **end for** features
(10) **end for** rules from *Beam*
(11) $Beam \leftarrow New\_Beam$
(12) **end while**

Fig. 1. Heuristic beam search rule construction algorithm for subgroup discovery.

3. Experimental results

In this section, we present results obtained by applying the subgroup discovery methodology in two gene expression problem domains.

- The first is the problem of distinguishing between acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML) described in [20]. Here, a training set with 38 samples (27 of type ALL and 11 of type AML) and a test set with 34 samples (20 of type ALL and 14 of type AML) have been available. Every sample is described by expression values of 7129 genes.
- The second domain is the multi-class cancer diagnosis problem for 14 different cancer types described in [45]. It has 144 samples in the training set and 54 samples in the test set. Every sample is described by expression values of 16,063 genes, where the first 7129 genes are the same as in the leukemia problem.

Training and test data sets, together with the description files, can be downloaded from http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi. Given the shortage of examples in gene expression problem domains, some sources suggest to use the so-called permutation test to assess the classifier accuracy, rather than isolating an independent test set. It has been shown in [23], however, that this alternative does not bring an advantage over the traditional accuracy estimation to which we thus adhere. Also note that a common technique known as leave-one-out cross-validation [21] would be a natural assessment choice when examples are rare. However, we have chosen to use the train/test data splits provided by [20] and [45], respectively, to be able to fairly compare our results with theirs.

Subgroup discovery starts from the available training sets with pre-computed presence call values. As described in Section 2.3, feature set F is very simple: it consists only of features $Attribute = A$ (gene expression absent) and $Attribute = P$ (gene expression present) generated for non-constant attributes. Features covering fewer than $min\_tp = |P|/2$ target class examples or fewer than $min\_tn = \sqrt{|N|}$ non-target class examples as well as all relatively irrelevant features have been eliminated in preprocessing of the SD algorithm. The user selected constants for the SD algorithm have been $min\_support = min\_tp = |P|/2$ and $beam\_width = 100$. Selection of these four constants is not critical: any *beam_width* value larger than 100 and any *min_support*, *min_tp*, and *min_tn* value up to 50% lower than the mentioned values result in the induction of same subgroups. The only observable consequence is the increase of the SD algorithm execution time.

3.1. The AML/ALL leukemia domain

For the first domain, 2844 attributes have been detected as totally irrelevant. After the elimination of absolutely and relatively irrelevant features, 639 relevant features remained when ALL is the target class and 622 when AML is the target class. For all generalization parameter values in the range 0.1–50 the SD algorithm has in both cases consistently constructed the same best subgroups shown in Table 3.

Table 4 presents the prediction results measured on the training set, independent test set, and on an independent test set consisting of leukumia samples from the

Table 3
Rules induced for the leukemia domain for classes acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL)

Models for the AML/ALL leukemia domain
AML class ← (LEPR_leptin_receptor EXPRESSED) AND (glutathione_s_transferase_microsomal EXPRESSED) ALL class ← (DF_D_component_of_complement_(adipsin) NOT EXPRESSED) AND (liver_mRNA_interferon_gamma_inducing_factor NOT EXPRESSED)

To improve the interpretability of induced models, gene values *P* and *A* have been replaced by EXPRESSED and NOT EXPRESSED, respectively.

Table 4
Sensitivity and specificity values for the leukemia training and test set, as well as for the leukemia samples from the multi class problem

Cancer	Training set		Test set			Leukemia from multi-class domain		
	Sens.	Spec.	Sens.	Spec.	Precision (%)	Sens.	Spec.	Precision (%)
AML	11/11	27/27	9/14	18/20	82	7/10	19/20	87
ALL	26/27	11/11	19/20	13/14	95	8/20	10/10	100

cancer multi-class domain. It can be noticed that although the rules have good sensitivity and specificity values[5] on the training set, the measured prediction quality on the test sets is not as good. Especially sensitivity values are not satisfactory because they are as low as 40% for the ALL rule tested on the multi-class leukemia test set. Interestingly, measured specificity values are much better and the lowest value 90% is measured for the AML rule on the two-class test set. Compared to the results reported in [20] our figures are slightly lower than those obtained by a weighted voting approach. Differences between results obtained on the training set and measured on the test sets, as well as differences among results obtained on different test sets indicate that some overfitting took place in spite of the implemented techniques of hypothesis space restriction. Although the rules are not that good for diagnostic purposes, i.e., distinguishing between AML and ALL disease types, induced rules describe some relevant—although smaller than expected—subgroups of disease classes. Their most significant quality is their easy interpretability as shown by the expert evaluation in Section 4.1.

3.2. The multi-class cancer domain

The same procedure was repeated for the second domain. For each cancer type as the target class, a rule (subgroup description) was constructed so that all other cancer types were treated as non-target class examples. In preprocessing, 2000 out of 16,063 attributes were detected as totally irrelevant. After the elimination of absolutely and relatively irrelevant features, for different classes 3300–8500 relevant features remained which in average presents 28% of all constructed features. The SD algorithm was used for all classes with the generalization parameter *g* equal to 5. Table 5 presents the sensitivity and the specificity results for the training set and the available independent test set. Additionally, the measured precision value is computed for the test set.

From Table 5, an interesting and important relationship between prediction results on the test set and the number of target class examples in the training set can be noticed. The obtained prediction quality on the test set is very low for many classes, significantly lower than those reported in [45]. For 7 out of 14 classes the measured precision is 0%. However, there are very large differences among the results for various classes (diseases). It can be noticed that the precision on the test set higher than 50% has been obtained for only 5 out of 14 classes. There are only three classes (lymphoma, leukemia, and CNS) with more than eight training samples and for all of them the induced rules have high precision on the test set, while for only 2 out of 11 classes with eight training cases (colorectal and mesothelioma) a high precision has been achieved. The classification properties corresponding to classes with 16 and 24 target class examples are comparable to the performances reported for these classes in [45], yet achieved by predictors much simpler than in the mentioned work. Consequently, we select those for expert interpretation in Section 4.2.

The results indicate that there is a certain threshold on the number of available training examples below which the subgroup discovery algorithm SD is not appropriate because it can not prevent overfitting despite the techniques designed for this purpose.[6] However, it seems that for only slightly larger training sets

[5] In Tables 4 and 5, sensitivity and specificity values are presented as fractions with the denominators presenting the numbers of positive and negative examples on which the rule quality has been tested.

[6] Improved results could be achieved by voting of best rules induced in several runs of the SD algorithm within the DMS covering algorithm; this work is out of the scope of this paper, as voting of numerous classifiers would hinder the interpretability of induced descriptions.

Table 5
Prediction results measured for 14 cancer types in the multi-class domain

Cancer	Training set		Test set		
	Sens.	Spec.	Sens.	Spec.	Precision (%)
Breast	5/8	136/136	0/4	49/50	0
Prostate	7/8	136/136	0/6	45/48	0
Lung	7/8	136/136	1/4	47/50	25
Colorectal	7/8	136/136	4/4	49/50	80
Lymphoma	16/16	128/128	5/6	48/48	100
Bladder	7/8	136/136	0/3	49/51	0
Melanoma	5/8	136/136	0/2	50/52	0
Uterus_adeno	7/8	136/136	1/2	49/52	25
Leukemia	23/24	120/120	4/6	47/48	80
Renal	7/8	136/136	0/3	48/51	0
Pancreas	7/8	136/136	0/3	45/51	0
Ovary	7/8	136/136	0/4	47/50	0
Mesothelioma	7/8	136/136	3/3	51/51	100
CNS	16/16	128/128	3/4	50/50	100

it can effectively detect relevant relationships. This conclusion is very optimistic because we can expect significantly larger gene expression databases to become available in the near future.

Table 6 presents the rules for the three cancer types with 16 and 24 training samples. Expert analysis of these rules in presented in Section 4.2. For all three diseases, a very good agreement in prediction results for the training and the test set can be noticed, which indicates that no significant training data overfitting has occurred. The sensitivity values measured on the test set are between 66 and 83% while the specificity value is always excellent and equal or almost equal to 100%, for all the three rules.

3.3. Comparing classification performance to previous results

We now relate the predictive classification performance of the five selected rules (2 for the AML/ALL domain and 3 for the multi-class domain) obtained by subgroup discovery to predictors for the corresponding classes previously reported in [20,45,10]. Due to differences in the presentation of predictive performance indicators in the mentioned literature, we convert them into a unified quantity of predictive accuracy, defined as the proportion of correctly classified testing instances among all testing instances. Further, we observe the number of genes (attributes) employed in the individual predictors.

The results for the AML/ALL domain are summarized in Table 7. We now describe the way we calculated the classification accuracy. For the subgroup discovery algorithm, we have two rules, each obtained by viewing one of the classes as the target class. To assess predictive accuracy, the two rules can be combined into a single two-class classifier, for example by a voting mechanism supplemented by a majority-class vote for instances not complying to the conditions of any of the rules. Such a

Table 7
The AML/ALL domain: comparing predictive accuracy and the number of involved genes for predictors obtained by the subgroup discovery approach (SD), by support vector machine construction (SVM), and by voting of informative genes [20]

Cancer	Classifier	Accuracy (%)	No. of genes
AML	SD	79.41	2
	SVM I [10]	88.24	50
	Voting [20]	93.94	50
ALL	SD	94.11	2
	SVM II [10]	94.11	50
	SVM III [10]	97.05	50

The three versions of SVM correspond to three groups of predictors with different parameterizations and different gene sets employed, as reported by [10].

Table 6
Rules induced for the multi-class cancer domain for cancer types with 16 (lymphoma and CNS) and 24 (leukemia) target class samples

Models for the multi-class cancer domain
lymphoma class ←
(CD20_receptor EXPRESSED) AND (phosphatidyl-inositol_3_kinase_regulatory_alpha_subunit NOT EXPRESSED)
leukemia class ←
(KIAA0128_gene EXPRESSED) AND (prostaglandin_d2_synthase_gene NOT EXPRESSED)
CNS class ←
(fetus_brain_mRNA_for_membrane_glycoprotein_M6 EXPRESSED) AND (CRMP1_collapsin_response_mediator_protein_1 EXPRESSED)

classifier would however no longer satisfy our requirement of simplicity. Therefore, we rather view each of the two rules as an individual binary classifier, interpreted under the closed-world assumption. That is, if the AML rule antecedent is not satisfied, then ALL is considered as the predicted class. The inverse principle is applied for the ALL rule. Each of the two rules is thus assigned its own accuracy value.

To calculate the predictive accuracy of the voting approach in [20], we consider that their predictor provided a class decision in 29 of the 34 testing cases and this decision was always correct. For the five undecided cases, we consider the accuracy of the majority vote on the test set ($\frac{20}{34} = 58.82\%$). The overall accuracy value is thus calculated as $\frac{29}{34}100\% + \frac{5}{34}58.82\% = 93.94\%$.

Finally, the SVMs in [10] provides a binary decision for all testing examples, therefore the accuracy calculation is straightforward using the provided counts of (in)correct classifications (Table 3 in [10]).

Two further notes have to be made on the numbers of genes (attributes) involved in the respective predictors, as quoted in Table 7. For the voting approach, Golub et al. [20] report that the number of correct decisions was maintained at 100% when reducing the number of employed genes to as low as 10. However, it is not reported for how many cases the reduced classifier was able to provide a decision. Therefore, we could not calculate the corresponding predictive accuracy value in the manner described above. For the SVMs in [10], the performance was also measured with the number of employed genes decreasing to as low as 4, and it was shown to fall rather quickly (see Table 2 in [10]). For a 4-best-genes attribute subset, the accuracy ranged from 54 to 93%, depending on the feature ranking method and the SVM parameterization. Even these rather low accuracies reported for the simplified classifiers probably overestimate the accuracies on the corresponding test sets. This is because they were measured by the cross-validation procedure combining the training and test sets for induction purposes and leaving only one example for testing at each validation stage. Thus, the assessed predictors were constructed from larger training sets than the predictors listed in Table 7.

Table 8 compares the predictive performance of the selected best classifiers obtained by the subgroup discovery approach in the multi-class cancer domain, with the SVM predictors for the corresponding classes achieved by [45] (see Fig. 4 of their paper). The accuracy for each class is measured for a binary classification task where all the examples of the given class are treated as positive and all the other examples as negative. Again, we observe the number of genes employed in the respective classifiers. Ramaswamy et al. [45] also investigate whether the predictive accuracy is sensitive to a decreasing number of available attributes (see Fig. 5 in [45]), but we cannot use these results as they are averaged over all classes and individual class results are not reported. However, the fact that the average accuracy of SVM falls from about 74% for 10,000 genes to about 57% for 3 genes indicates that—unlike with the SD approach—satisfactory accuracy can not be expected from SVM with a very small number of employed attributes.

Table 8
Multi-class cancer domain: Comparing predictive accuracy and number of involved genes for selected predictors obtained by the subgroup discovery approach (SD) and by support vector machine (SVM) construction [45]

Class	Classifier	Accuracy (%)	No. of genes
Lymphoma	SD	98.14	2
	SVM [45]	100.00	16063
Leukemia	SD	94.44	2
	SVM [45]	98.14	16063
CNS	SD	98.14	2
	SVM [45]	100.00	16063

4. Expert evaluation of induced models

This section provides an expert evaluation of the induced subgroup descriptions by one of the authors (J.T.), who was not involved in the rule discovery process described above. To make the text accessible to a reader without biological background, we provide a less formal explanation of certain terms. In general, albeit a simplified view, cellular processes which increase proliferation (cellular division) and inhibit apoptosis (cellular death) are consistent with a phenotype of cancerous cell.

4.1. The AML/ALL domain

Cancers which originate from hematopoietic (blood) cells are called leukemias and lymphomas. Acute leukemias can be of either lymphoid[7] origin (acute lymphocytic leukemia, ALL) or myeloid[7] origin (acute myelogenous leukemia, AML).

The best-scoring rule for the AML disease class was the following:

AML class:
(LEPR_leptin_receptor EXPRESSED) AND (glutathione_s_transferase_microsomal EXPRESSED)

The first condition assumes the expression of the leptin receptor. The obesity gene product leptin regulates food intake, but it is also important in the regulation of inflammation, immunity and hematopoiesis[8] [16]. The leptin receptor, a single transmembrane-spanning molecule, is a member of the cytokine[9] receptor super-

[7] A subclass of white blood cells.
[8] Blood forming.
[9] A secreted signaling peptide (protein).

family. It is expressed on the hematopoietic stem cells [22], and, while absent from samples of ALL, it is frequently expressed in primary and secondary AML [32]. Interaction of leptin with its receptor has proliferative and anti-apoptotic effect on AML blasts [32]. Leptin, secreted from bone marrow adipocytes,[10] stimulates both myeloid[7] development and bone marrow angiogenesis.[11] Furthermore, it has been shown [24] that inhibition of the leptin receptor signaling by anti-leptin receptor antibody decreased both microvessel formation and number of AML blasts in the bone marrow.

Regarding the second condition, Glutathion-*S*-transferases (GST) are liver cytosolic[12] and microsomal[12] enzymes, which metabolize toxic substances [28]. Detoxification of toxic substances (e.g., environmental mutagens and chemotherapeutic drugs used in cancer treatment) is important for both the development of malignancies and their response to treatment. Whereas the condition regards the microsomal kind of GST, most of existing literature and knowledge is concerned with the cytosolic kind. For example, it is known that cytosolic GST are polymorphic in humans and null variants of some GST isoforms seem to increase oxidative stress on hematopoietic stem cells. This may in turn lead to a higher incidence of leukemia or chemotherapy resistant disease with poorer outcome [53,27,33].

Concerning the possible leptin–GST interaction, it is remarkable that in a study [4] of experimental hepatotoxicity with reduced cytosomal GST in a murine model, exogenous administration of leptin resulted in decreased detoxification and high levels of reactive byproducts. Again, our model assumes the *elevated* expression of *microsomal* GST and thereby does not directly parallel the mentioned study. However, it suggests that the leptin receptor and GST may form a combined factor relevant to pathophysiology of AML, and along with [4] it motivates a further investigation of the possible interaction of leptin with the GST family.

The best-scoring rule for the ALL disease class was the following:

ALL class:
(DF_D_component_of_complement_(adipsin) NOT EXPRESSED) AND (liver_mRNA_interferon_gamma_inducing_factor EXPRESSED)

The first condition is concerned with adipsin, also termed complement factor[13] D. This is an enzyme producing the acylation[14]-stimulating protein (ASP), which increases triglyceride synthesis in adipocytes[10] [9]. Adipsin is expressed in cell lines derived from human monocytes [5], hepatocytes[15] [6], astroglioma[16] [7] and gastric cancer [30], but not—to our knowledge—in ALL. Significantly, a recent analysis of ALL and AML transcription profiling data identified adipsin as one out of three best targets for investigating the basic biology of ALL/AML and their mutual distinction [10]. Our model thus confirms this basic observation of [10], but is more informative in that it specifically articulates the absence of adipsin expression assumed for the ALL class.

Interferon-gamma-inducing factor, assumed to be expressed by the second condition, was discovered in 1995 [43] and later termed interleukin-18 (IL-18). IL-18 is secreted by activated macrophages[7] and induces high levels of interferon-gamma production in T cells[17] [49]. The rule's assumption is compatible with the previous study [52] where increased IL-18 expression has been correlated with ALL. The expression has been correlated also with cutaneous[18] natural killer lymphoma [3], cutaneous T-cell lymphoma [3], metastatic breast cancer [37], lymphohistiocytosis[19] [51], and high risk AML [57]. It has been suggested that IL-18 could lead to antitumor effects in some cancers through induction of apoptosis [42], and that IL-18 is likely involved in the autonomy of leukemic cells [58].

A remark should be made concerning the mutual relationship of the two genes involved in the rule. Interferon-gamma has been observed in the human astroglioma[15] cell line to stimulate the expression of complement factors[13] B and C2, closely related to adipsin (complement factor D) [7]. Adipsin itself, however, was refractory to the IL-18 stimulation. This is in agreement with the simultaneous presence of IL-18 and absence of adipsin as stipulated by the rule.

4.2. The multi-class cancer domain

Here we discuss the discovered rules for three respective cancer classes: lymphoma, leukemia, and central nervous system (CNS) cancers. As we have mentioned already, leukemias and lymphomas are cancers originating from hematopoietic (blood) cells.

[10] "Fat cells".
[11] Forming of vessels.
[12] The adjectives cytosolic and microsomal refer to two different locations within the cell.
[13] Substance present in blood that plays role in blood-clotting and immune response.
[14] Addition of a carbohydrate group.
[15] Liver cells.
[16] Brain cancer.
[17] Thymus-derived white blood cell. The thymus is a gland responsible for maturation of some immune cells.
[18] Skin-related.
[19] Disease associated with high numbers of histiocytes (macrophages).

The following rule was found for the lymphoma class:

Lymphoma class:
(CD20_receptor EXPRESSED) AND (phosphatidylinositol_3_kinase_regulatory_alpha_subunit NOT EXPRESSED)

The first condition stipulates the expression of the CD20 receptor. CD20 receptor, a calcium channel,[20] is a lineage-specific[21] B-cell[22] antigen present on lymphoid cells. CD20 lymphoid marker is used routinely in diagnosis of lymphomas. The identification of this gene is thus reassuring and confirms that our search strategy is able to detect genes already known to be characteristic of specific malignancy such as lymphoma.

Phosphatidyl-inositol-3-kinase (PI3K), assumed not expressed by the second condition, is a key molecule in intracellular signaling. It transmits signals from the cellular membrane to the nucleus, and its activation leads to cytokine[9] production and cell division. PI3K is also critical for killing (cytotoxicity) of tumor cells by T cells[17] and natural killer (NK) cells[7] [59]. Therefore the absence of PI3K activation may compromise immune surveillance and result in environment permissive for malignant growth. While PI3K is a necessary for survival of some leukemic cells [55,1], it is conceivable that in other malignancies, presumably driven by different proliferation signals, the absence of PI3K (with or without dysregulation of T cell and NK surveillance) could result in clonal proliferation and lymphoma [48,59].

For the leukemia class, we have the following rule:

Leukemia class:
(KIAA0128_gene EXPRESSED) AND (prostaglandin_d2_synthase_gene NOT EXPRESSED)

KIAA0128 gene (Septin 6), addressed by the first condition, is a member of a family of filament-forming[23] proteins, septins, forming heteropolymer complexes involved in cytoskeletal organization and cell division. Septin 6 has been identified as a fusion partner of the MLL gene in infants with acute leukemias [8,44,47,17]. The MLL gene is frequently rearranged and fused to partner genes in ALL and AML. Out of more than 40 gene fusion partners of MLL gene identified to-date, three are septins, and the AML type with the Septin 6 (KIAA0128 gene)—MLL fusion likely represents a subset on infant AML with common leukemogenesis pathway [29].

The second condition is concerned with the absence prostaglandin D synthase (PGDS) expression. PGDS is an enzyme active in the production of prostaglandins (pro-inflammatory an anti-inflammatory molecules). Elevated expression of PGDS has been found in brain tumors, ovarian and breast cancer [50,26], while hematopoietic PGDS has not been, to our knowledge, associated with leukemias.

Viewing the rule as a whole, the absence of PGDS expression may be a part of the "molecular signature" reflecting either the general tissue type (leukocytes) or the specific, KIAA0128 (Septin 6) dependent, leukemic process. Future studies should determine whether the identification of Septin 6 is due to frequent Septin 6—MLL rearrangements in our series or whether the Septin 6 expression is associated with other types of leukemia as well. Collectively, these observations could lead to a more general role for Septin 6 in leukemias with and without MLL rearrangements.

Lastly, we address the rule found for the CNS class.

CNS class:
(fetus_brain_mRNA_for_membrane_glycoprotein_M6 EXPRESSED) AND (CRMP1_collapsin_response_mediator_protein_1 EXPRESSED)

Concerning the first condition, the membrane glycoprotein M6 functions as a neuron-specific[24] calcium channel. Upon nerve growth factor stimulation the M6 protein appears to promote neuronal differentiation [41], and the antibodies against M6 affect the survival of cerebellar[25] neurons [56].

As for the second condition, members of the collapsin/semaphorin family,[26] including collapsin response mediator protein 1, CRMP1, play an important role in proliferation,[27] and pathfinding of growing axons to reach their targets in nervous system [12,13]. Both M6 and CRMP1 appear to have multifunctional roles in shaping neuronal networks, and their function as survival (M6) and proliferation (CRMP1) signals may be relevant to growth promotion and malignancy.

5. Conclusions

This study aimed to test the feasibility of inducing simple, rule-based models for gene expression data. We argue that a major advantage of such models is their direct interpretability by domain experts. The prediction results obtained on independent test sets as well expert

[20] A molecule on the surface of a cell membrane that facilitates the inflow and outflow of calcium.
[21] Specific for a particular development path from a stem cell to a differentiated (specialized) cell.
[22] Bone marrow (B) derived white blood cell.
[23] Forming a cellular "skeleton."
[24] That is, it only operates in nerve cells.
[25] A specific area of the brain.
[26] A family of proteins regulating the growth of neurons.
[27] Growth.

analysis of induced rules demonstrate that the chosen approach, based on the presented subgroup discovery methodology, can be a useful tool for the detection of relevant relationships between sample classes (diseases) and measured gene expression values. In contrast to other machine learning applications for gene expression data analysis, we have started from presence call (categorical) values. Features based on presence call values are very easy for human interpretation and this significantly contributes to rules being accepted as comprehensible disease models.

The interpretation of the subgroup discovery results yields several biological observations: out of the five best-scoring rules (for five respective problems) selected for expert evaluation, two (lymphoma and leukemia classes) are judged as reassuring and three (AML, ALL, and CNS classes) have a plausible, albeit partially speculative explanation. Namely, the best-scoring rule for the lymphoma class in the multi-class cancer recognition problem (containing 16,063 attributes) contains a feature corresponding to a gene routinely used as a marker in diagnosis of lymphomas (CD20), while the other part of the conjunction (the PI3K gene) seems to be a plausible biological co-factor. The best-scoring rule for the leukemia class contains a gene whose relation to the disease is directly explicable (Septin 6). In the problem of distinguishing AML from ALL, the best-scoring rule related to the AML class connects in a logical conjunction two genes, GST and leptin (out of 7192 original genes), whose co-activity was previously under biological investigation in a model of impaired detoxification, and supports a possibility that they may form a combined factor relevant to the etiology of AML.

In spite of the number of findings in agreement with the bio-medical state-of-the-art, discovery of known factors in the considered malignancies was not the ultimate goal of this study. The main goal of the methodology is the discovery of unknown and never thought-off relationships, in a form instantly understandable to an expert. Such relationships can in turn be tested and potentially validated by means of current rapidly advancing bio-medical research and, later, clinical trials.

Although the subgroup discovery approach emphasizes a strong restriction of the hypothesis space with the intention to prevent data overfitting, the results demonstrate that this phenomenon, linked to the overwhelming number of existing feature combinations in the attribute-rich domain, can not be completely eliminated, especially in domains and target classes with a small number of samples. Therefore, for several target classes (with fewer than 16 positive examples) we have not been able to induce a well-generalizing rule submittable to expert interpretation. It is promising, however, that the obtained prediction quality of the induced rules grows very rapidly with the increased size of the training set and we expect to have significantly larger gene expression domains in the near future from which it will be possible to induce comprehensible, highly reliable, and highly predictive disease models. This will help in disease prediction and classification, and in attempts to better understand the biology of malignancy, to risk stratify cancer patients and, in future applications, to implement treatment strategies targeted at individual patients.

Acknowledgments

This work was supported by the Croatian Ministry of Science, Education and Sport, the Slovenian Ministry of Education, Science and Sport, and the Czech Ministry of Education through the project MSM 212300013.

References

[1] Abbott RT, Tripp S, Perkins SL, Elenitoba-Johnson KS, Lim MS. Analysis of the PI-3-kinase-PTEN-AKT pathway in human lymphoma and leukemia using a cell line microarray. Mod Pathol 2003;16(6):607–12.

[2] Agrawal R, Imielinski T, Shrikant R. Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD conference on management of data, Washington, DC; 1993. p. 207–16.

[3] Amo Y, Ohta Y, Hamada Y, Katsuoka K. Serum levels of interleukin-18 are increased in patients with cutaneous T-cell lymphoma and cutaneous natural killer-cell lymphoma. Br J Dermatol 2001;145(4):674–6.

[4] Balasubramaniyan V, Kalaivani SJ, Nalini N. Role of leptin on alcohol-induced oxidative stress in Swiss mice. Pharmacol Res 2003;47(3):211–6.

[5] Barnum SR, Volanakis JE. In vitro biosynthesis of complement protein D by U937 cells. J Immunol 1985;134(3):1799–803.

[6] Barnum SR, Volanakis JE. Biosynthesis of complement protein D by HepG2 cells: a comparison of D produced by HepG2 cells, U937 cells and blood monocytes. Eur J Immunol 1985;15(11):1148–51.

[7] Barnum SR, Ishii Y, Agrawal A, Volanakis JE. Production and interferon-gamma-mediated regulation of complement component C2 and factors B and D by the astroglioma cell line U105-MG. Biochem J 1992;287(Pt 2):595–601.

[8] Borkhardt A, Teigler-Schlegel A, Fuchs U, Keller C, Konig M, Harbott J, et al. An ins(X;11)(q24;q23) fuses the MLL and the Septin 6/KIAA0128 gene in an infant with AML-M2. Genes Chromos Cancer 2001;32(1):82–8.

[9] Cianflone K, Xia Z, Chen LY. Critical review of acylation-stimulating protein physiology in humans and rodents. Biochim Biophys Acta 2003;1609(2):127–43.

[10] Chow ML, Moler EJ, Mian IS. Identifying marker genes in transcription profiling data using a mixture of feature relevance experts. Physiol Genom 2001;3(5):99–111.

[11] Clark P, Niblett T. The CN2 induction algorithm. Machine Learn 1989;3(4):261–83.

[12] Cohen-Salmon M, Crozet F, Rebillard G, Petit C. Cloning and characterization of the mouse collapsin response mediator protein-1, Crmp1. Mamm Genome 1997;8(5):349–51.

[13] Deo RC, Schmidt EF, Elhabazi A, Togashi H, Burley SK, Strittmatter SM. Structural bases for CRMP function in plexin-dependent semaphorin3A signaling. EMBO J 2004;23:9–22.

[14] Domingos P. The role of Occam's razor in knowledge discovery. Data Mining and Knowledge Discovery 1999;3:409–25.
[15] Dudoit S, Fridlyand J, Speed T. Comparison of discrimination methods for the classification of tumors using gene expression data. Tech Report 576, University of California, Berkeley http://stat-www.berkeley.edu/sandrine/tecrep/576.pdf; 2000.
[16] Fantuzzi G, Faggioni R. Leptin in the regulation of immunity, inflammation, and hematopoiesis. J Leuk Biol 2000;68:437–46.
[17] Fu JF, Liang DC, Yang CP, Hsu JJ, Shih LY. Molecular analysis of t(X;11)(q24;q23) in an infant with AML-M4. Genes Chromos Cancer 2003;38(3):253–9.
[18] Fürnkranz J. Separate-and-conquer rule learning. Artif Intell Rev 1999;13:3–54.
[19] Gamberger D, Lavrač N. Expert-guided subgroup discovery: Methodology and application. J Artif Intell Res 2002;17:501–27.
[20] Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 1999;286:531–7.
[21] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning, data mining, inference and prediction. Berlin: Springer; 2001.
[22] Hino M, Nakao T, Yamane T, Ohta K, Takubo T, Tatsumi N. Leptin receptor and leukemia. Leuk Lymph 2000;36(5–6):457–61.
[23] Hsing T, Attoor S, Dougherty E. Relation between permutation-test P values and classifier error estimates. Machine Learn Eraing, Special Issue on Machine Learning in the Genomics 2003;52:11–30.
[24] Iversen PO, Drevon CA, Reseland JE. Prevention of leptin binding to its receptor suppresses rat leukemic cell growth by inhibiting angiogenesis. Blood 2002;100:4123–8.
[25] Jovanoski V, Lavrač N. Classification rule learning with APRI-ORI-C. In: Progress in artificial intelligence: proceedings of the tenth portuguese conference on artificial intelligence. Berlin: Springer; 2001. p. 44–51.
[26] Kawashima M, Suzuki SO, Yamashima T, Fukui M, Iwaki T. Prostaglandin D synthase (beta-trace) in meningeal hemangiopericytoma. Mod Pathol 2001;14(3):197–201.
[27] Kearns PR, Chrzanowska-Lightowlers ZMA, Pieters R, Veerman A, Hall AG. Mu class glutathione *S*-transferase mRNA isoform expression in acute lymphoblastic leukaemia. Br J Haematol 2003;120(1):80–8.
[28] Kelner MJ, Stokely MN, Stovall NE, Montoya MA. Structural organization of the human microsomal glutathione *S*-transferase gene (GST12). Genomics 1996;36(1):100–3.
[29] Kim HJ, Ki CS, Park Q, Koo HH, Yoo KH, Kim EJ, et al. MLL/SEPTIN6 chimeric transcript from inv ins(X;11)(q24;q23q13) in acute monocytic leukemia: report of a case and review of the literature. Genes Chromos Cancer 2003;38(1):8–12.
[30] Kitano E, Kitamura H. Synthesis of factor D by gastric cancer-derived cell lines. Int Immunopharmacol 2002;2(6):843–8.
[31] Klsgen W. Explora: a multipattern and multistrategy discovery assistant advances in knowledge discovery and data mining. Cambridge: MIT Press; 1996.
[32] Konopleva M, Mikhail A, Estrov Z, Zhao S, Harris D, Sanchez-Williams G, et al. Expression and function of leptin receptor isoforms in myeloid leukemia and myelodysplastic syndromes: proliferative and anti-apoptotic activities. Blood 1999;93:1668–76.
[33] Krajinovic M, Labuda D, Sinnett D. Glutathione *S*-transferase P1 genetic polymorphisms and susceptibility to childhood acute lymphoblastic leukaemia. Pharmacogenetics 2002;12(8):655–8.
[34] Lavrač N, Gamberger D, Turney P. A relevancy filter for constructive induction. IEEE Intell Syst Their Appl 1997;13:50–6.
[35] Liu H, Motoda H. Feature selection for knowledge discovery and data mining. Dodretch: Kluwer Academic Publishers; 1998.
[36] Li J, Wong L. Geography of differences between two classes of data. In: Proceedings of the sixth european conference on principles of data mining and knowledge discovery. Berlin: Springer; 2002. p. 325–37.
[37] Merendino RA, Gangemi S, Ruello A, Bene A, Losi E, Lonbardo G, et al. Serum levels of interleukin-18 and sICAM-1 in patients affected by breast cancer: preliminary considerations. Int J Biol Markers 2001;16(2):126–9.
[38] Michalski RS, Mozetič I, Hong J, Lavrač N. The multi-purpose incremental learning system AQ15 and its testing application on three medical domains. In: Proceedings of the fifth national conference on artificial intelligence. Los Altos, CA: Morgan Kaufmann; 1986. p. 1041–5.
[39] Mitchell T. Machine learning. New York: McGraw-Hill; 1997.
[40] Molla M, Waddell M, Page D, Shavlik J. Using machine learning to design and interpret gene-expression microarrays. AI Mag, Special Issue on Bioinformatics 2004:23–44.
[41] Mukobata S, Hibino T, Sugiyama A, Urano Y, Inatomi A, Kanai Y, et al. M6a acts as a nerve growth factor-gated Ca(2+) channel in neuronal differentiation. Biochem Biophys Res Commun 2002;297(4):722–8.
[42] Ohtsuki T, Micallef MJ, Kohno K, Tanimoto T, Ikeda M, Kurimoto M. Interleukin 18 enhances Fas ligand expression and induces apoptosis in Fas-expressing human myelomonocytic KG-1 cells. Anticancer Res 1997;17(5A):3253–8.
[43] Okamura H, Tsutsi H, Komatsu T, Yutsudo M, Hakura A, Tanimoto T, et al. Cloning of a new cytokine that induces IFN-gamma production by T cells. Nature 1995;378(6552):88–91.
[44] Ono R, Taki T, Taketani T, Kawaguchi H, Taniwaki M, Okamura T, et al. SEPTIN6, a human homologue to mouse Septin6, is fused to MLL in infant acute myeloid leukemia with complex chromosomal abnormalities involving 11q23 and Xq24. Cancer Res 2002;62(2):333–7.
[45] Ramaswamy S, Tamayo P, Rifkin R, Mukherjee S, Yeang CH, Angelo M, et al. Multiclass cancer diagnosis using tumor gene expression signatures. Proc Natl Acad Sci USA 2001;98(26):15149–54.
[46] Schaffer C. Overfitting avoidance as bias. Machine Learning 1993;10:153–78.
[47] Slater DJ, Hilgenfeld E, Rappaport EF, Shah N, Meek RG, Williams WR, et al. MLL-SEPTIN6 fusion recurs in novel translocation of chromosomes 3, X, and 11 in infant acute myelomonocytic leukaemia and in t(X;11) in infant acute myeloid leukaemia, and MLL genomic breakpoint in complex MLL-SEPTIN6 rearrangement is a DNA topoisomerase II cleavage site. Oncogene 2002;21(30):4706–14.
[48] Stankovi T, Stewart GS, Byrd P, Fegan C, Moss PA, Taylor AM. ATM mutations in sporadic lymphoid tumours. Leuk Lymph 2002;43(8):1563–71.
[49] Steele TA. Chemotherapy-induced immunosuppression and reconstitution of immune function. Leuk Res 2002;26(4):411–4.
[50] Su B, Guan M, Zhao R, Lu Y. Expression of prostaglandin D synthase in ovarian cancer. Clin Chem Lab Med 2001;39(12):1198–203.
[51] Takada H, Ohga S, Mizuno Y, Suminoe A, Matsuzaki A, Ihara K, et al. Oversecretion of IL-18 in haemophagocytic lymphohistiocytosis: a novel marker of disease activity. Br J Haematol 1999;106(1):182–9.
[52] Taniguchi M, Nagaoka K, Kunikata T, Kayano T, Yamauchi H, Nakamura S, et al. Characterization of anti-human interleukin-18 (IL-18)/interferon-gamma-inducing factor (IGIF) monoclonal antibodies and their application in the measurement

of human IL-18 by ELISA. J Immunol Methods 1997;206(1–2): 107–113.

[53] Voso MT, D'Alo F, Putzulu R, Mele L, Scardocci A, Chiusolo P, et al. Negative prognostic value of glutathione *S*-transferase (GSTM1 and GSTT1) deletions in adult acute myeloid leukemia. Blood 2002;100:2703–7.

[54] Wrobel S. An algorithm for multi-relational discovery of subgroups. In: Proceedings of the first european symposium on principles of data mining and knowledge discovery. Berlin: Springer; 1997. p. 78–87.

[55] Xu Q, Simpson S-E, Scialla TJ, Bagg A, Carroll M. Survival of acute myeloid leukemia cells requires PI3 kinase activation. Blood 2003;102:972–80.

[56] Yan Y, Lagenaur C, Narayanan V. Molecular cloning of M6: identification of a PLP/DM20 gene family. Neuron 1993;11(3):423–31.

[57] Zhang B, Wang Y, Zheng GG, Ma XT, Li G, Zhang FK, et al. Clinical significance of IL-18 gene over-expression in AML. Leuk Res 2002;26(10):887–92.

[58] Zhang B, Ma XT, Zheng GG, Li G, Rao Q, Wu KF. Expression of IL-18 and its receptor in human leukemia cells. Leuk Res 2003;27(9):813–22.

[59] Zhong B, Liu JH, Gilvary DL, Jiang K, Kasuga M, Ritchey CA, et al. Functional role of phosphatidylinositol 3-kinase in direct tumor lysis by human natural killer cells. Immunobiology 2002;205(1):74–94.