Memorial
University of Newfoundland

Technical Report #2007-03
Department of Computer Science
Memorial University of Newfoundland
St. John’s, NL, Canada

# GENETIC PROGRAMMING BASED DNA MICROARRAY ANALYSIS FOR CLASSIFICATION OF CANCER

by

Michael Rosskopf (1), Heiko A. Schmidt (2), Udo Feldkamp (3), Wolfgang Banzhaf (4)

[1]Department of Computer Science, University of Dusseldorf, Germany,
Email: rosskopf@cs.uni-duesseldorf.de

[2]Center for Integr. Bioinformatics Vienna, Vienna, Austria,
Email: heiko.schmidt@univie.ac.at

[3]Department of Computer Science, University of Dortmund, Germany,
Email: feldkamp@cs.uni-dortmund.de

[4]Department of Computer Science, Memorial University of Newfoundland
St. John’s, NL, A1B 3X5, Canada, Email: banzhaf@cs.mun.ca

Department of Computer Science
Memorial University of Newfoundland
St. John’s, NF, Canada A1B 3X5

November 2007

# Genetic Programming based DNA Microarray Analysis for Classification of Cancer

Michael Rosskopf (1), Heiko A. Schmidt (2),
Udo Feldkamp (3), Wolfgang Banzhaf (4)

(1) Dept. of Computer Science, University of Düsseldorf, Germany, rosskopf@cs.uni-duesseldorf.de
(2) Center for Integr. Bioinformatics Vienna, Vienna, Austria, heiko.schmidt@univie.ac.at
(3) Dept. of Computer Science, University of Dortmund, Germany, feldkamp@cs.uni-dortmund.de
(4) Dept. of Computer Science, Memorial Univ. of Newfoundland, Canada, banzhaf@cs.mun.ca

**Abstract.** In this study the advantages of statistical gene selection are combined with the power of Genetic Programming (GP) to build classifiers for assigning gene expression microarray data samples to categories characteristic of certain cell states. To that end we implemented different statistical measures in a program called GENEACTIVATOR and tested their applicability to gene selection. Subsequently we used the general purpose GP-system DISCIPULUS to train classifiers. We applied our approach to four different human cancer gene expression datasets publicly available, including multi-class sets. The results indicate that using gene selection and GP as implemented in DISCIPULUS is an appropriate method for gene expression data analysis.

## 1 Introduction

DNA microarrays provide insight into gene expression levels in cells [4, 38]. They can be used for cancer diagnosis, since gene expression patterns in tumor tissue differ from those in healthy tissue, reflecting the sets of genes active in the different tissue types. Because thousands of genes can be examined simultaneously with microarrays, new opportunities arise by analyzing these patterns to reveal cancer types and their state of progress enabling better treatment. Even new tumor classes might be discovered by analyzing data gained from microarrays.

The key problem of evaluation of gene expression data is to find patterns in the apparently unrelated values measured. With increasing numbers of genes spotted on microarrays visual inspection of these data has become impossible and, hence, the importance of computer analysis has substantially increased in recent years. Well-studied datasets of different phenotypes are publicly available to train and evaluate supervised pattern analysis algorithms for classification and diagnosis of unknown samples.

Such datasets are tables where the samples are arranged in columns and the expression values of genes are represented in rows. In this article the data are represented as $n \times m$ matrices where $n$ is the number of samples and $m$ is the number of genes. So for each sample an $m$-dimensional vector of expression values is available, one entry for each gene $i$ with $1 \leq i \leq m$. For each gene, on the other hand, there exists an $n$-dimensional vector with one expression value for each sample $j$ with $1 \leq j \leq n$. Furthermore, each dataset contains an additional vector of size $n$ which holds the class label of samples describing their phenotype, e.g., benign or malignant status.

A gene expression profile obtained from a microarray is simply a snapshot of the current state of the tissue under study reflecting expression intensity of genes controlled by processes

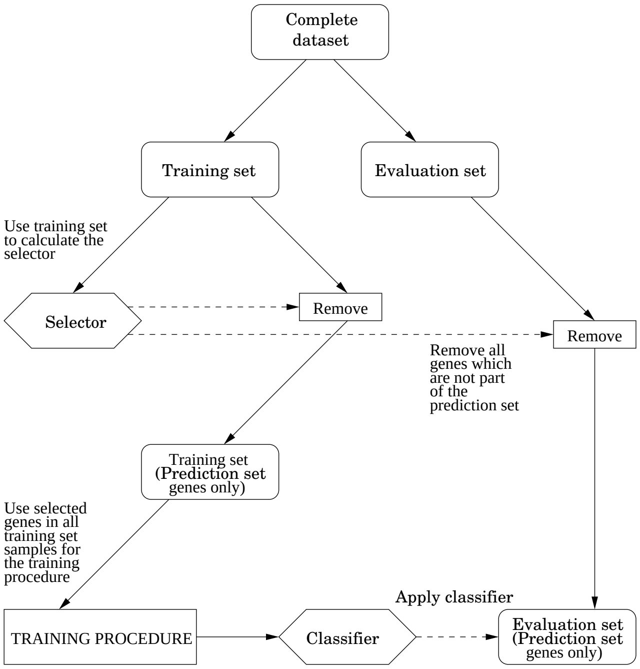
Complete dataset
Training set
Evaluation set
Use training set to calculate the selector
Selector
Remove
Remove
Remove all genes which are not part of the prediction set
Training set (Prediction set genes only)
Use selected genes in all training set samples for the training procedure
Apply classifier
TRAINING PROCEDURE
Classifier
Evaluation set (Prediction set genes only)

**Fig. 1.** General approach for classification of gene expression data. After an initial separation of data into a training and an evaluation set, the selector algorithm determines relevant genes and the classifier (after running the training algorithm on it) produces a prediction of both seen and unseen samples.

like cell cycle, metabolic pathways, state or age of the cell. Only some of the genes examined will be usually related to disorders like cancer or other diseases. Thus, the set of relevant genes has to be extracted from the entire set of genes and often approaches are used similar to the one illustrated in Figure 1: The complete set of genes is first reduced by a *selector* to a so-called *prediction set* of relevant genes for diagnosis. Hence, a selector has to be generated that selects only those genes which show high correlation with the phenotype of interest. Often, the statistical criterion of the signal-to-noise ratio has been implemented in selector algorithms. Subsequently, only the selected genes are used to build *classifiers* for diagnosis. Such classifiers have been created by techniques based on $k$-nearest neighbors [32], hierarchical clustering [1], bi-clustering [11], weighted voting [20], support vector machines [10, 21], or Bayesian networks [17, 23], to name a few prominent ones. Usually classifiers are evaluated on a separate subset of data which consists of samples not used for training.

Evolutionary Algorithms (EA) have been used for solving problems of both selection and prediction in microarray analysis. GAs have been employed for building selectors [13, 28–31]. Each component of the representation corresponds to one gene and the state of the component denotes whether the gene is selected or not.

GP [5, 25] on the other hand, as the most complex form of EAs, has been shown to work well for recognition of structures in large data sets. GP has been applied to microarray data to generate programs that reliably predict the health/malignancy states of tissue, or which type of cancer is present in the tissue. [34] used GP to classify microarray data introducing a Symbolic Discriminant Analysis (SDA). SDA avoids the disadvantage of linear discriminant analysis, where the discriminant functions have to be specified. In their approach the selection of genes and discriminant functions is performed by GP automatically. The system has been used for analysing microarray data of human leukemia. The same authors used GP to identify autoimmune disease based on microarray data [35]. They also modified SDA to test the consistency of leave one out cross-validation (LOOCV) for the selection of genes. In LOOCV one sample is removed from the dataset. The other samples are used to generate a classifier which is applied to classify the sample removed. The process is iterated for all samples and results are averaged. Moore used the selection rate during LOOCV as a fitness value for the relevance of each gene. Relevant genes were then turned over to a second SDA classifying the samples (reviewed in [33]). [37] extended this approach with more complex SDA functions.

A different form of GP with programs as Boolean expression rules was used for analyzing 6 functional classes of yeast genes by [19] and a 4-class cancer set by [14]. Tree GP was used by [26] to identify embryonal tumors of the central nervous system. LOOCV was performed 10 times for each partitioning and the impact of every gene was measured like in [35]. In [22] a rule-based approach was used similar to [14] and [19], but mixed with statistical gene selection to analyze lymphoma data. In that contribution the GP system used the 30 genes with the best signal-to-noise ratio.

In this article we combine statistical gene selection and a GP system for classification. This two-step procedure increases the speed of training, since all genes not showing strong correlation with the examined phenotypes are removed. Reducing the number of genes helps to avoid the negative effects of overfitting. Here we tested several different statistical measures on their applicability for gene selection. We applied this approach to four different sets of cancer data containing between 2 and 14 different tissue classes.

## 2 Material and Methods

We perform a two-step procedure consisting of (i) a gene selection step and (ii) a training and classification step. The selection step is implemented in a program called GENEACTIVATOR while training of classifiers is performed with DISCIPULUS [15, 16]. The data flow is shown in Figure 2. The full dataset is divided into two subsets: The *training set* and the *evaluation set*. The former is used for gene selection with GENEACTIVATOR to create a reduced training set which is fed into DISCIPULUS for training. The latter is kept separate to evaluate the results of our approach. For technical reasons a reduced evaluation set is produced containing only those genes selected by GENEACTIVATOR based on the training set. This set is employed for evaluating the classifier, but it is neither utilized by the gene selection process nor by the training procedure of DISCIPULUS.

### 2.1 Gene selection

GENEACTIVATOR selects genes with a high relevance from the training set to discriminate between phenotypes. To this end, different procedures were used to determine the relevance of a gene. Every gene selection procedure calculates a relevance score for all $m$ genes. Genes with the highest scores are selected to generate a new reduced matrix. Seven

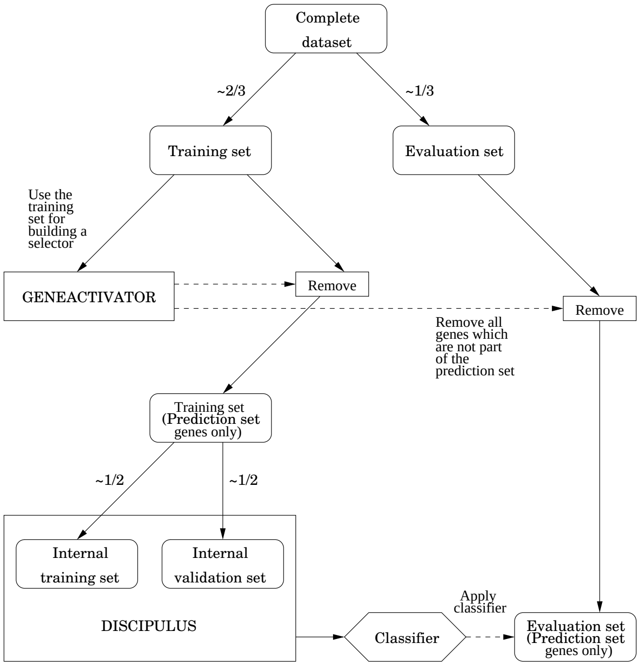
Complete dataset
~2/3
~1/3
Training set
Evaluation set
Use the training set for building a selector
GENEACTIVATOR
Remove
Remove
Remove all genes which are not part of the prediction set
Training set (Prediction set genes only)
~1/2
~1/2
Internal training set
Internal validation set
DISCIPULUS
Apply classifier
Classifier
Evaluation set (Prediction set genes only)

**Fig. 2.** Data flow of the approach studied here. Selection of relevant genes is done by GENEACTIVATOR, training of classifiers is done by the GP tool DISCIPULUS, which uses an internal training and validation set.

different selection procedures are evaluated here to compare their relative performance on selecting relevant genes:

1. calculating the interval range (IR) of all $n$ expression levels of a gene,
2. the standard deviation (SD) of all $n$ values,
3. a two-partition (2P, see below) criterion,
4. the mean difference (MD) between the expression values of both classes,
5. the signal-to-noise ratio (S2N) (c.f. [36]),
6. the Fisher criterion (FC) [7] and
7. a cluster count criterion (CC, see below).

Methods 1-3 are unsupervised, while methods 4-7 are supervised using the class labels of samples for the selection step. GENEACTIVATOR is used for binary problems where the set of samples can be separated into class-0 and class-1 samples. In Section 2.3 the handling of data with more than two classes is explained.

The criteria 2P and CC are newly introduced in this article. The following formula is used to calculate the 2P relevance of a gene $x$:

$$R_{2P}(x) = \frac{\mu_+(x) - \mu_-(x)}{\sigma_+(x) + \sigma_-(x)}, \tag{1}$$

where $\mu_+(x)$ is the mean of all expression levels larger than the overall-mean of all $n$ expression levels, while $\mu_-(x)$ is the mean of all expression levels smaller the overall-mean. Values

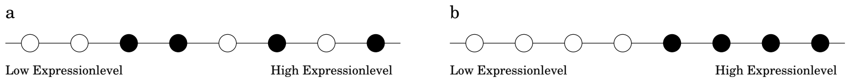

**Fig. 3.** Two examples for a sorted set of samples from genes with low (left) and high (right) relevance scores with respect to the cluster count criterion: Each circle stands for the expression level of a particular gene in a sample, sorted from low to high expression levels. Black and white indicate class labels, 0 and 1. Gene *a* has a score of 2, gene *b* has the best possible score (for 8 samples) of 6.

**Table 1.** Function set of DISCIPULUS.

| Simple arithmetic operations | Complex operations |
|---|---|
| Addition | Root |
| Subtraction | Trigonometric |
| Multiplication | Comparison |
| Division | Condition |
| | Data transfer (between registers) |

$\sigma_+(x)$ and $\sigma_-(x)$ are the corresponding standard deviations. $R_{2P}(x)$ measures the ability of gene $x$ to divide the $n$ values into two clusters.

$R_{CC}(x)$ can be computed with the following algorithm :

```
1. Sort all $n$ expression values of $x$ according to their expression level. Let $x_i$ be the expression level of $x$ in sample $i$ after sorting and let $c(x_i)$ be the class label of sample $i$ after sorting.
2. LET $s = 0$
3. FOR $i = 1$ TO $n - 1$:
   IF $c(x_i) = c(x_{i+1})$ THEN $s = s + 1$
4. RETURN $s$
```

The variable $s$ is used as the relevance score $R_{CC}(x)$ for gene $x$. If $s$ is high, the gene is assumed to have a high ability to differentiate the two classes, since the expression levels form bigger clusters, whereas a gene with a low value of $s$ contains more small clusters. Examples are given in Figure 3. $s$ should be normalized by the number of samples in the set if different sets are to be compared.

All genes are ordered by their relevance scores $R(x)$ and those with the highest relevance scores are selected. If not stated otherwise, we always select the 20 most relevant genes. It is our experience that this choice provides a good tradeoff between optimization time, accuracy and overfitting of data. Different numbers of genes showed inferior results (data not shown here).

### 2.2 The GP system DISCIPULUS

DISCIPULUS is a general purpose GP tool [15, 16] which can be used for regression and binary classification problems. The software uses GP to breed a population of small programs able to, in this case, classify the microarray data. Therefore, we refer to the programs created as classifiers.

**Table 2.** Example classifier program evolved with DISCIPULUS, containing addition, substraction, multiplication, division and power.

```
L0: f[0] = f[0] + v[2];
L1: f[0] = f[0] - v[1];
L2: f[0] = f[0] / v[8];
L3: f[0] = f[0] - v[6];
L4: f[0] = pow(2,f[0]);
L5: f[0] = f[0] * v[2];
```

**Representation of individuals** A DISCIPULUS classifier is very similar to an assembler program consisting of simple arithmetic operations, comparisons, and conditions on register data. Some more complex functions are available, too (see Table 1). Classifiers can be conveniently converted into C- or JAVA-code to be used in other programs. Table 2 shows a short example of a classifier program. Calculation register `f[0]` is initialized with 0. `v[i]` denotes the expression level of gene $i$ in the current sample, where $i$ is the index of the gene selected. In this example classifier, expression levels of genes 1, 2, 6 and 8 are used with mathematical operations. The result in `f[0]` is returned and used for class prediction: If the value in `f[0]` is smaller than the threshold of 0.5, the sample analyzed is predicted to be of class-0, and of class-1 otherwise.

**Training procedure** DISCIPULUS implements the special training procedure illustrated in Figure 4. The entire optimization process is separated into a series of independent optimization processes, called runs. The 30 best classifiers of all runs are collected allowing not more than 5 to be added per run. Performing many independent runs with varying initial parameter settings (see below) increases the probability to find appropriate classifiers, since GP is a heuristic method. The training set can be split into an internal training set and an internal validation set with the former being used for training while both being applied for scoring the 30 best classifiers. Although this feature can reduce overfitting, for a small number of samples in the dataset, this option will reduce the information utilized for training due to an exclusion of the validation set from the actual training.

The number of correctly classified samples (the hit rate) from the training set is used as primary fitness criterion. In the event of a tie, the distance between threshold and the numerical value is used as a second fitness criterion. Tournament selection reduces 4 to 2 classifiers, which are subsequently subjected to crossover and mutation.

In addition to single-program classification, DISCIPULUS also provides teams of classifiers. Classifiers with best results are collected into a team with one vote for each member. The class with the majority of votes is selected as the team answer. It has been shown that a team is frequently better than any particular member of the team [8]. We used teams with nine individuals each for multiclass prediction shown in the section on experiments below.

**Parameter settings** DISCIPULUS provides a large number of parameters. Since the optimal parameter setting is typically not known, parameters are initialized for each single run with a randomized value around default values at the outset. Later, in order to increase the probability of choosing good values, default values are updated with the parameters of

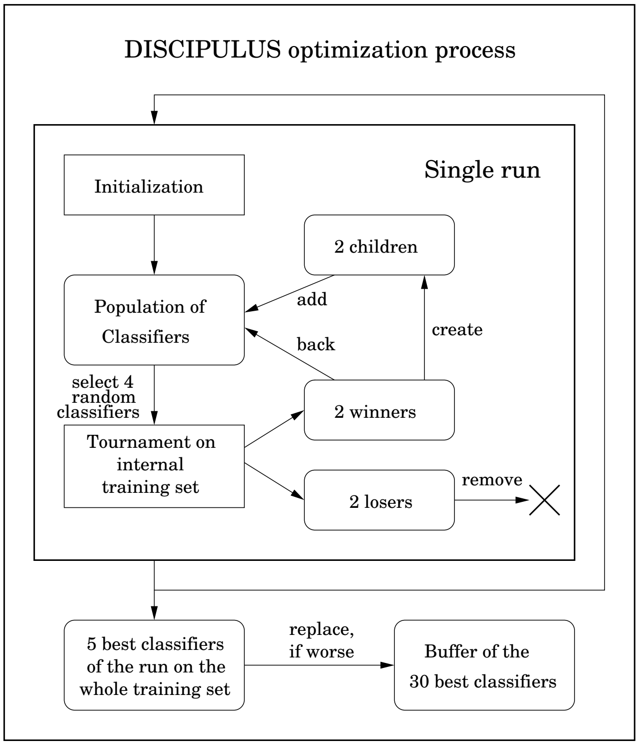
DISCIPULUS optimization process
Single run
Initialization
2 children
Population of Classifiers
add
back
create
select 4 random classifiers
2 winners
Tournament on internal training set
2 losers
remove
5 best classifiers of the run on the whole training set
replace, if worse
Buffer of the 30 best classifiers

**Fig. 4.** Optimization process of DISCIPULUS.

**Table 3.** Mean values of GP parameters. Randomization was used to assign the actual parameters (see text).

| Parameter | Value |
|---|---|
| Population size | 500 |
| Mutation frequency | 95% |
| Recombination frequency | 50 % |
| Max program size | 512 bytes |

the best runs so far, thus implementing a racing algorithm (described in [16]). Important parameter values are given in Table 3. If not stated otherwise, we used these default values.

**Class weights** If one class is represented by only a small fraction of samples from the training set, it is recommended to assign different weights to the classes. Otherwise the selective pressure to learn features of the smaller class might be too low. Let $h_0$ be the rate of correct predictions on the class-0-samples and let $h_1$ be the rate for class-1-samples accordingly. These hit-rates are modified by the weights $w_0$ and $w_1$ (quantified below) to get a weighted hit-rate:

$$\text{Weighted Hit-Rate} = (h_0 \cdot w_0) + (h_1 \cdot w_1) \tag{2}$$

**Table 4.** Comparison of datasets used. The last two columns show the number of samples in the training and evaluation set. Notice that there is no explicit evaluation set for colon data provided.

| Set | #Classes | #Genes | #Training set | #Evaluation set |
|---|---|---|---|---|
| COLON | 2 | 2000 | 62 | - |
| ALL/AML | 2 | 7129 | 38 | 34 |
| SRBCT | 4 | 6567 | 63 | 25 |
| GCM | 14 | 16063 | 144 | 54 |

The resulting weighted hit-rate is used to measure the fitness of a classifier.

### 2.3 Dealing with multi-class sets

Both GENEACTIVATOR and DISCIPULUS were designed to handle binary classification problems. Hence we have implemented a one-versus-rest (OVR) method (c.f. [36]) for multiclass problems. To that end we constructed binary classifiers for each class $i$, that can distinguish this class $i$ from the rest of the samples. At first all relevant genes were selected by GENEACTIVATOR for each class $i$ with class labels for all samples of class $i$ set to 1, for all remaining samples set to 0. This process was iterated for each class and DISCIPULUS was applied to all selected sets separately.

With increasing number of classes, however, the ratio of positive samples decreases. As explained above this decreases the selective pressure on classifiers to recognize positive samples while training, which leads to poor hit-rates. In order to compensate for this effect we used the following class weights: Let $\Phi$ be a dataset and let $\Phi_i$ be the subset of all samples belonging to class $i$. The weight of class $i$ is calculated by:

$$w(\Phi_i) = \frac{100}{2\,|\Phi_i|} \tag{3}$$

Hence, if a classifier for class $i$ is constructed, $w_1 = w(\Phi_i)$ (positive for class $i$) and the weight of all remaining samples, which do not belong to class $i$, is $w_0 = 50/(|\Phi| - |\Phi_i|)$ corresponding to Equation 2. With these weights, the importance of positive and negative samples is equal. The weights can be interpreted as percentages, so that both classes have 50% impact on the fitness calculation.

In order to predict the class of a sample, classifiers for all classes are applied to that sample. Contradictory results, e.g., two classifiers returning a positive output, were treated by a simple post-processing procedure. The best teams of size nine for each class were analyzed for the number of team members making a positive prediction. The class with most positive votes wins the classification. If more than one class have maximum number of positive votes the sample is considered as ‘undecidable’. In the event that no class garners more than one vote the sample is classified as ‘miscellaneous’.

### 2.4 Datasets

We examined four different publicly available datasets of cancer tissues (c.f. Table 4). Two of these are binary class datasets: A colon set with healthy and malignant colon tissues by [1] and a set with two subtypes of leukemia (ALL/AML) by [20]. The remaining two data sets are multiclass datasets, (i) a set of four types of small round blue cell tumors (SRBCT)

**Table 5.** Results on the two-class datasets, Colon and ALL/AML: Columns show hit-rates of the best classifiers on the corresponding evaluation set (mean hit-rates of the 30 best classifiers). The results of cross-validation are shown in the LOOCV columns. LOOCV results are percentages of correct predictions over all 30 best classifiers of each sample left out. The numbers of samples that had at least 15 out of 30 classifiers with incorrect prediction are given in brackets.

| | Colon | | ALL/AML | |
|---|---|---|---|---|
| Selector | Hit-rate | LOOCV | Hit-rate | LOOCV |
| IR | 77.42 (64.62) | 79.68 (12) | 64.71 (51.77) | 61.67 (12) |
| SD | 74.19 (63.66) | 78.76 (11) | 79.41 (68.33) | 79.47 (7) |
| 2P | 83.87 (73.01) | 80.16 (9) | 97.06 (89.41) | 85.00 (6) |
| MD | 87.10 (77.31) | 79.78 (10) | 94.12 (89.41) | 92.11 (2) |
| S2N | 90.32 (77.74) | 80.22 (11) | 97.06 (89.71) | 90.53 (1) |
| FC | 90.32 (77.10) | 80.05 (10) | 94.12 (82.25) | 91.84 (1) |
| CC | 77.42 (67.63) | 83.60 (8) | 97.06 (83.24) | 88.51 (2) |

[24] and (ii) the GCM set with 14 different classes of tumors described in [36]. As shown in Table 4 these sets contain between 2,000 and 16,063 genes, but only a small number of samples (38 to 144). For ALL/AML, SRBCT, and the GCM set the partitioning into training and evaluation set is described in the publications referenced. In order to compare our results we used the original partitioning. For the colon data no partitioning has been described, hence, we assigned samples randomly to a training and an evaluation set with 31 samples each, with the constraint that both classes are represented with an equal number of samples in both sets. A particularity is found in the SRBCT evaluation set which includes five samples not belonging to any of the four classes in the training set. Thus, a test can be made for how the system handles a totally unknown class.

## 3 Experiments and Results

### 3.1 Binary data

The primary goal of experiments with binary datasets was to evaluate the different gene selection methods. We applied all seven methods to both binary datasets and selected the best 20 genes of each. Each analysis comprised 300 independent GP runs, each being terminated after 300 generations of fitness stagnation.

The training set was assigned to both DISCIPULUS internal training and validation sets allowing to use the full information present in the original set. After training the resulting 30 classifiers where applied to the evaluation sets. Table 5 presents the results for the Colon and ALL/AML datasets. Hit-rates and mean hit-rates are listed for the best classifiers. The results show that for all selection strategies except IR the method performed better on the ALL/AML data with a hit-rate of up to 97.06%, while on the Colon data we were only able to achieve up to 90.32%.

We compared these results with leave oneout cross-validation (LOOCV) to measure the methods ability to generalize. For each sample left out, a complete series of 100 GP runs was performed to classify the sample, with a fixed number of 100 generations per single run. The results are also shown in Table 5.

Altogether, 2P is the best unsupervised method on both datasets. On the ALL/AML set 2P is even better than most of the supervised methods. Only S2N is slightly better on

**Table 6.** Example classifiers for the four classes of the SRBCT dataset generated with DISCIPULUS. `GENE` *i* stands for the expression level of gene *i*. Hit-rate is 100% on the evaluation set, i.e. this is a perfect solution. Gene numbers correspond to the order in the original dataset.

```
IF((GENE 123 * GENE 165) > 0.5)
        THEN 'Burkitt lymphoma'
IF((((GENE 1319 / GENE 2050) + 0.54216) * 0.22590) > 0.5)
        THEN 'Ewing family of tumors'
IF((GENE 742 * GENE 255 * 0.34336) > 0.5)
        THEN 'Neuroblastoma'
IF((GENE 147 * GENE 187 * GENE 2047) > 0.5)
        THEN 'Rhabdomyosarcoma'
```

average for all 30 best classifiers. 2P failed to detect 6 samples in LOOCV with a majority of classifiers. Here the supervised methods were clearly better (1 or 2 failures). Surprisingly, the very simple methods IR and SD also reached a hit-rate of almost 80% in some cases. But many genes with high 2P or S2N scores also have a large range and standard deviation, so that there is a large intersection in the prediction sets selected by these different methods. Among the supervised methods MD seems to be relatively good in comparison to the widely used S2N method. Through division by standard-deviations, S2N produces a kind of normalized MD. It seems that GP achieves only a slight advantage through this normalization. FC is similar to MD and S2N and produced similar results here. The CC approach is based upon an idea different from MD, S2N and FC. Overall it seems that it offers no real advantage in comparison to the other methods. An exception is the result from LOOCV on the Colon dataset, where CC provided the best results.

### 3.2 Multiple class data

After examination of the binary datasets, we analyzed multiple class cancer sets. Here, our main goal was to find good classifiers, with accurate predictions for as many samples as possible. We had to choose one selector method over the others, due to the increased computational effort with multiple classes.

On binary datasets 2P proved to be the best unsupervised selection method. But for multiple classes this selector produced only poor results (not shown) and therefore not suitable for multiclass prediction. The supervised methods MD, S2N, and FC all produced similar results on the binary sets, with S2N slightly better than the others. CC performed suboptimal in most cases except LOOCV on Colon data. So here we only used the S2N criterion for the more complex multiclass analysis. For training we ran GP 300 times for each class in a dataset (OVR), with a run being terminated after 300 generations of fitness stagnation.

We start with the multiclass analysis of the SRBCT data. After training classification rates of 100% for the evaluation samples were achieved by using the post-processing algorithm described in section 2.3. In particular, the five samples not belonging to any of the four classes were correctly classified as unknown. We also trained classifiers using a prediction set described in [14]. In that study a perfect classifier with only 10 genes is described. It was generated by a special variant of GP. Our classifier trained on the same 10 genes, also achieved a perfect hit-rate of 100%.

**Table 7.** Comparison of best results on the different evaluation datasets used in this contribution. Second column: Best results (hit rate of best single individual on the evaluation dataset) using the GENEACTIVATOR/DISCIPULUS combination. Third and fourth columns: Results of other studies (cited). See text for further discussion.

| Data | Best results here | Other EA results | Other methods |
|---|---|---|---|
| COLON | 90.32% | 100.00% [13] | 90.32% [18] |
| | | 90.00% [30] | 87.09% [1] |
| ALL/AML | 97.06% | 100.00% [13] | 97.06% [2] |
| | | 97.06% [28] | 97.06% [12] |
| | | 97.06% [30] | 97.06% [27] |
| | | | 94.11% [40] |
| | | | 94.11% [18] |
| | | | 85.29% [20] |
| SRBCT | 100.00% | 100.00% [14] | 100.00% [27] |
| | | | 100.00% [39] |
| | | | 100.00% [40] |
| GCM | 62.96% | 84.30% [13] | 80.00% [3] |
| | | | 78.00% [36] |
| | | | 76.60% [39] |

Using the aforementioned post-processing algorithm for the GCM set hit-rates of only 46.29% were achieved on the evaluation set. By increasing the number of genes selected for each of the 14 OVR optimizations from 20 to 50 the algorithm obtained a hit-rate of 62.96%. In [36] improved hit-rates were reached with larger sets of genes. Prediction sets with 20 genes only might be too small for the GCM problem.

The same dataset has been analyzed by [13]. They enlarged their training set by incorporating the correct hit-rate on the evaluation set (in their case as a further Pareto goal). Applying the evaluation set in DISCIPULUS we could also achieve a classification rate of 100.00%, again with the post-processing algorithm and 50 genes in the prediction set. It must be emphasized that the evaluation set in our case was not used for training during the runs, but only for evaluating classifiers after training in order to sort through them on the basis of their generalization performance.

The same method was tested on the SRBCT set to find small classifiers which could classify all samples. Overall we found a large number of perfect classifiers for this dataset. Table 6 shows an example of a classifier predicting each class correctly for all samples in the training and in the evaluation set.

## 4 Discussion

Table 7 shows a comparison of our results with those described in the literature. Prediction set sizes used in these studies vary considerably. A fair comparison is therefore difficult to achieve. Our results on the two binary sets are as strong as those achieved by others, except the results by [13]. Their results, however, were only obtained by adding the evaluation set into training the classifiers. Generally, the results on the Colon dataset are weaker than those on the ALL/AML set. It seems that there is more noise in the Colon data than in the ALL/AML data. In multi-class classification we reached a hit-rate of 100.00% on the SRBCT dataset, again as strong as other studies. On the more complex 14-class GCM dataset recognition rates were relatively weak compared to other studies, suggesting further research is necessary.

Altogether it has been confirmed that GP is a suitable method for gene expression data analysis. Performance strongly depends on the gene selection method and the quality of datasets. It should be possible to advance the accuracy of our method further, for instance by a pre-analysis of the data to choose a suitable gene selection method. If the signal-to-noise ratio is poor, the unsupervised 2P method should be used in two-class problems. It is not clear why the genes found by [14] are so beneficial to generate a classifier. Only about half of them have a good signal-to-noise ratio. So the search for better selection procedures is important, particularly for multivariate procedures ranking combinations of genes. Gene expression values are usually not independent ¿from each other, due to the dependencies in metabolic pathways. Hence, for classification purposes one gene alone might be useless, but in combination with other genes it might be significant for a reliable diagnosis.

A number of features to improve the power of DISCIPULUS would be desirable:

– Better support for non-binary problems, perhaps with automatic OVR or the automatic use of All Pairs (AP).[1]
– Implementation of LOOCV.
– Batch analysis from command line for high-throughput analysis.

With the continuous advance in data collection and data management methods used with gene expression data [9, 6] we are going to see more demand for potent analysis tools in the future. We hope to be able to conduct broader studies in the future that will confirm the validity of our approach.

## Acknowledgement

DISCIPULUS was developed by Register Machine Learning Technologies, Littleton CO (www.aimlearning.com). The authors would like to thank Register Machine Learning Technologies for allowing us to use DISCIPULUS (Academic) Version 3.0 and Markus Conrads for software support.

GENEACTIVATOR is freely available from the following website:

http://www.cs.mun.ca/~banzhaf/gene_activator/

Data sets and best classfiers are available upon request.

## References

1. U. Alon, N. Barkai, D.-A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.-J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, 96: 6745–6750, 1999.
2. V. Aris and M. Recce. A method to improve detection of diseases using selectivly expressed genes in microarray data. In S. M. Lin and K. F. Johnson, editors, *Proceedings of CAMDA '00*, 69–81, 2002. Kluwer Academic, Norwell, MA.

[1] In AP, one class is distinguished from only one other class and classifiers have to be trained for all pairs of classes (c.f. [36])

3. A.-M. Bagirov, B. Ferguson, S. Ivkovic, G. Saunders, and J. Yearwood. New algorithms for multi-class cancer diagnosis using tumor gene expression signatures. *Bioinformatics*, 19: 1800–1807, 2003.
4. P. Baldi and G.-W. Hatfield. *DNA Microarrays and Gene Expression.* Cambridge University Press, Cambridge, U.K., 2002.
5. W. Banzhaf, P. Nordin, R. Keller, and F. Francone. *Genetic Programming - An Introduction.* Morgan Kaufmann, San Francisco, 1998.
6. T. Barrett, T.O. Suzek, D.B. Troup, S.E. Wilhite, W.C. Ngau, P. Ledoux, D. Rudnev, A.E. Lash, W. Fujibuchi, R. Edgar. NCBI GEO: mining millions of expression profiles — database and tools. *Nucl Acid Res*, 33: D562–D566, 2005.
7. C.-M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, U.K., 1995.
8. M. Brameier and W Banzhaf. Evolving Teams of Predictors with Linear Genetic Programming. *Genetic Programming and Evolvable Machines*, 2: 381–407, 2001
9. A. Brazma, P. Hingamp, J. Quackenbush, et al.. Minimum information about microarray experiment (MIAME) – Toward standards for microarray data. *Nature Genetics*, 29: 365–372, 2001.
10. M.P.S. Brown, W.-N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, M. Ares Jr, D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *PNAS*, 97: 262–267, 2000.
11. S. Busygin, G. Jacobsen, and E. Krämer. Double conjugated clustering applied to leukemia microarray data. In *2nd SIAM ICDM, Workshop on clustering high dimensional data*, 2002.
12. S. Busygin, O.A. Prokopyev, and P.M. Pardalos. Feature Selection for Consistent Biclustering via Fractional 0-1 Programming. *J Combinatorial Optimization*, 10: 7–21, 2005.
13. K. Deb and A.-R. Reedy. Classification of two and multi-class cancer data reliably using multi-objective evolutionary algorithms. *Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur,* KanGAL Report Number 2002006, 2003.
14. J.-A. Driscoll, B. Worzel, and D. MacLean. Classification of gene expression data with genetic programming. In R.L. Riolo, editor, *Genetic Programming: Theory and Practice*. Kluwer, Norvell, MA, 2003.
15. J.-A. Foster. Review: Discipulus: A commercial genetic programming system. *Genetic Programming and Evolvable Machines*, 2: 201–203, 2001.
16. F.-D. Francone. *Discipulus Owner's Manual.* Register Machine Learning Technologies, Littleton CO, www.aimlearning.com, 2001.
17. N. Friedman, M. Linial, I. Nachmann, and D. Peer. Using Bayesian Networks to Analyze Expression Data. *J. Computational Biology* 7:601-620, 2000.
18. T.-S. Furey, N. Cristianini, N. Duffy, and D.-W. Bednarski. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16: 906–914, 2000.
19. R.-J. Gilbert, J.-J. Rowland, and D.-B. Kell. Genomic computing: explanatory modelling for functional genomics. In D. Whitley, D. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, and H.-G. Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, 551–557, San Francisco, 2000. Morgan Kaufmann.
20. T.-R. Golub, D.-K. Slonim, P. Tamayo, and C. Huard. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286: 531–537, 1999.
21. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46: 389–422, 2002.
22. J.-H. Hong and S.-B. Cho. Lymphoma cancer classification using genetic programming with SNR features. In M. Keijzer et al., editors, *Proc. Genetic Programming 7th European Conference (EuroGP 2004)*, 78–88, Springer, Berlin, 2004.
23. K.-B. Hwang, D.-Y. Cho, S.-W. Park, S.-D. Kim, and B.-T. Zhang. Applying machine learning techniques to analysis of gene expression data: Cancer diagnosis. In S. M. Lin and K. F. Johnson, editors, *Proceedings of CAMDA '00*, 69–81, 2002. Kluwer Academic, Norwell, MA.

24. J. Khan, J.-S. Wei, M. Ringer, L.-H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.-R. Antonescu, C. Peterson, and P.-S. Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neutral networks. *Nature Medicine*, 7: 673–679, 2001.
25. J. Koza. *Genetic Programming.* MIT Press, Cambridge, MA, 1992.
26. W.-B. Langdon and B.-F. Buxton. Genetic programming for mining DNA chip data from cancer patients. *Genetic Programming and Evolvable Machines*, 5: 1–7, 2004.
27. Y. Lee and C.-K. Lee. Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics*, 19: 1132–1139, 2003.
28. L. Li, T.-A. Darden, C.-R. Weinberg, A.-J. Levine, and L.-G. Pedersen. Gene assessment and sample classification for gene expression data using a genetic algorithm/k−nearest neighbor method. *Combinatorial Chemistry & High Throughput Screening*, 4: 727–739, 2001.
29. L. Li, W. Jiang, X. Li, K.-K. Moser, Z. Guo, L. Du, Q. Wang, E.-J. Topol, Q. Wang, and S. Rao. A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset. *Genomics*, 85: 16–23, 2005.
30. J. Liu and H. Iba. Selecting informative genes using a multiobjective evolutionary algorithm. *CEC '02. Proceedings of the 2002 Congress on Evolutionary Computation, Honolulu, HI*, Vol 1: 297 – 302, IEEE Press, New York, 2002.
31. J.-J. Liu, G. Cutler, W. Li, Z. Pan, S. Peng, T. Hoey, L. Chen, and X.-B. Ling. Multiclass cancer classification and biomarker discovery using GA-based algorithms. *Bioinformatics*, 21: 2691–2697, 2005.
32. D. Michie, D.-J. Spiegelhalter, and C.-C. Taylor. *Machine learning, neural and statistical classification.* Prentice Hall, 1994.
33. J.-H. Moore. Cross validation consistency for the assessment of genetic programming results in microarray studies. In S. Cagnoni et al., editors, *Applications of Evolutionary Computing: EvoWorkshops 2003*, 99–106, Springer, Berlin, 2003.
34. J.-H. Moore, J.-S. Parker, and L.-W. Hahn. Symbolic discriminant analysis for mining gene expression patterns. In L. De Raedt and P. Flach, editors, *Lecture Notes in Artificial Intelligence 2167*, 372–381, Springer, Berlin, 2001.
35. J.-H. Moore, J.-S. Parker, N.-J. Olsen, and T.-M. Aune. Symbolic discriminant analysis of microarray data in autoimmune disease. *Genetic Epidemiology*, 23: 57–69, 2002.
36. S. Ramaswamy, P. Tamayo, R. Rifkin, and S. Mukherjee. Multiclass cancer diagnosis using tumor gene expression signatures. *PNAS*, 26: 15149–15154, 2001.
37. D.-M. Reif, B.-C. White, N. Olsen, T. Aune, and J.-H. Moore. Complex function sets improve symbolic discriminant analysis of microarray data. In E Cantú-Paz et al., editors, *Proc. Genetic and Evolutionary Computation Conf. (GECCO 2003)*, 2277–2287, Springer, Berlin, 2003.
38. T. Speed. *Statistical Analysis of Gene Expression Microarray Data.* Chapman and Hall/CRC, Boca Raton, London, New York, Washington D.C., 2003.
39. A. Statnikov, C.-F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21: 631–643, 2005.
40. R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *PNAS*, 99: 6567–6572, 2001.