Genetic Programming based DNA Microarray Analysis for Classification of Tumor Tissues

Michael Rosskopf* Udo Feldkamp† Wolfgang Banzhaf‡§

July 6, 2004

Abstract

Gene expression data gained from DNA-microarrays have been examined with several approaches during the last few years. In many cases statistical methods or learning techniques were used to classify the origin or state of tissues the gene samples have been taken from. In this article we present a Genetic Programming (GP) approach to this problem. We trained classifiers on four different public cancer data sets, including multi-class sets, using the general purpose GP-system DISCIPULUS. In a preprocessing step a subset of all sampled genes has to be selected to get smaller data sets and to avoid overfitting. We examined several different statistical measures on their applicability to this selection. The results indicate that GP is an appropriate method for gene expression data analysis.

1 Introduction

DNA microarrays are a powerful tool for cancer diagnosis (introductions to microarrays are [4] and [32]). Gene expression patterns in tumor tissue are different from those in healthy tissue, both in intensity and timing, reflecting different sets of active genes in both tissue types. Since many thousands of genes can be examined with DNA chips simultaneously, new possibilities arise through analyzing these patterns for finding out the degree of maliciousness and kind of cancer. Even new tumor classes might be uncovered by analyzing the data gained from DNA-chips.

A key problem, however, is the accurate evaluation of gene expression data, because it is difficult to find structure in the apparently connectionless values measured. Since visual inspection of these data has become unfeasible with rising spot density on DNA chips and large numbers of samples, the importance of computer analysis has grown in recent years. Well-known data are used to train pattern analysis algorithms which later should be able to classify unknown patterns properly. Data sets often contain correlated gene activities which are lumped together by clustering. After clustering has been applied, gene expression patterns can be classified and probability statements for unknown data can be made. Many different stochastic learning procedures, like k-nearest neighbors [29, 30], hierarchical clustering [1], self-organizing maps [20, 22], support vector machines [9, 8, 11, 18, 21, 25, 31] or Bayesian networks [17, 23], to name a few prominent ones, have already been applied in this context. By the same token Evolutionary Algorithms (EA) had been tested for solving this problem. EA is a generic term for machine learning methods, employing techniques of artificial evolution to improve solutions that should solve a specified problem. The concept of EAs comprises very many different approaches like Evolution Strategies (ES), Evolutionary Programming (EP), Genetic Algorithms (GA) [6] or Genetic Programming (GP) [24]. Li et al. used GAs for analyzing cancer data sets [27]. A method which applying multi-objective GAs to this problem was used by Deb and Ready [12, 13, 14]. In both works the evolutionary methods had been employed just for the selection of genes in order to get a good (and in the studys by Deb and Ready also a smallest) prediction set. After that the classification was done with statistical tests.

Here we examine whether GP is suitable for gene expression data analysis [5]. As is known from other areas, GP works well for recognition of structures in large data sets. The representation of

*Dept. of Computer Science, University of Düsseldorf, GERMANY
†Dept. of Computer Science, University of Dortmund, GERMANY
‡Dept. of Computer Science, Memorial University of Newfoundland, CANADA, banzhaf@cs.mun.ca
§To whom correspondence should be addressed

```
L0:   f[0]/=v[4];
L1:   f[0]+=v[2];
L2:   f[0]-=v[1];
L3:   f[0]-=v[8];
L4:   f[0]-=v[6];
```

Table 1: One of the best classifiers on the ALL/AML dataset.

solutions in GP are any programs in general case. The artificial evolution works by improving those programs. Here, we evolve programs to make reliable predictions as to whether an examined tissue is healthy or malignant, or which kind of cancer is present in the tissue at hand. At the outset of such an evolution, a population of programs is created randomly. The performance of the programs is tested on a training set of well-known data. Programs showing high performance are selected to produce offspring, thereby applying variation methods like mutation and recombination, i.e., changing program instructions and interchanging instruction blocks. This procedure is iterated until a stop criterion is met. More information about GP can be found in [5]. A restricted form of GP where the programs were actually boolean expressions consisting of arithmetic operations and a comparison was used for analyzing a 4-class cancer set in [15] with great results.

2 Materials and Methods

2.1 Datasets

We examined four different publicly available data sets of cancer tissues. Two data sets consist of two classes: The colon cancer set by Alon et al. ([1]) and the ALL/AML leukemia set by Golub et al. ([20]). The remaining two data sets were (i) the 4-class small round blue cell tumor (SRBCT) set ([33]) and (ii) the 14-class GCM set described in [31]. These sets contain between 2000 and 16063 genes, but only a small number of samples (62 to 144). Table 2 gives an overview of the sets.

The datasets are tables where the samples are arranged in the columns and the genes are represented by the rows. In this article a dataset is denoted as a $n \times m$ matrix where n is the number of samples and m is the number of genes. So a gene is a n-dimensional vector where the i-th component holds the expression value of this gene in the i-th sample for $1 \leq i \leq n$ and a sample is a m-dimensional vector where the j-th component holds the expression value of this sample in the j-th gene for $1 \leq j \leq m$.

2.2 The GP system DISCIPULUS

For our work we used the GP-system DISCIPULUS developed by Register Machine Learning Technologies Inc. [16]. DISCIPULUS is a general purpose GP-system which can be used for regression and binary classification problems. The software creates small programs with the technique of GP which should solve a defined question, for example to decide whether a specific sample is malignant or not. As we used the system only for this kind of classification problems, we call the generated programs classifiers. A classifier is very similar to an assembler program with commands for simple arithmetic operations, comparisons, conditions, data transfer and also trigonometric functions. In terms of GP these operations establish the function set. Classifiers can be simply converted to C- or JAVA-code for the use in other programs. Table 1 shows a short example of a classifier-program that was generated by using the ALL/AML data. This program makes predictions with a hit-rate of 100.00% on a test set unknown to the system (more about this in the result section). All registers `f` are initialized with 0 at the beginning. The values `v` are the inputs, i.e., the expression value of the i-th gene of a sample is stored in `v[i]` while testing the sample. The program returns `true` if the calculated value is greater than a defined threshold (here = 0.5) and `false` otherwise.

For the artificial evolution of the programs DISCIPULUS needs three subsets of the data. Thus for our GP runs, every data set has been divided into three subsets of roughly equal size, a training, a validation and an applied set. DISCIPULUS uses tournament selection [16] to compare the fitness of the programs on the training set. The validation set is used to determine how well the best programs generalize. The programs with the best results on both sets are taken for producing offspring. So both sets are used for training, where the validation set provides internal validation of generalization performance. This partition counters overfitting [16]. The applied ortest set is used to evaluate the generated classifier programs on data that the programs never have seen and could therefore not use to learn (external validation). These evaluations are the main results and they are presented in

Set	# Classes	# Genes	# Samples (All)	# Samples (Test set)
Colon	2	2000	62	20
ALL/AML	2	7129	72	24
SRBCT	4	6567	88	25
14-class set	14	16063	144	54

Table 2: Size comparison of datasets used.

section 3. In addition to single-program classification, DISCIPULUS also offers a team mode. The programs giving the best results are collected together in a team and every member of a team has one vote. To classify a sample the class that got most votes is selected.

In our data preparation we removed all extended information from the original sets keeping only gene expression levels. Because DISCIPULUS can handle only binary classification problems, we used the one-versus-all (OVA) method [31] for the two multiclass classifications. In this method, a separate classifier is trained for each class that treats the union of all other classes as one class of negative examples. When classifying an unknown sample, it is presented to all OVA classifiers. Obviously, postprocessing of the classifications is needed to get an unambiguous result, for example if more than one classifier returns a positive response.

2.3 Gene-selection with GENEACTIVATOR

In its present form, DISCIPULUS can handle a maximum of 63 features where one gene is one feature in our case. With this restriction preprocessing was necessary to reduce the number of genes. While this means loss of information from the original datasets, a high number of genes is disadvantageous, because a small number of samples in relation to the number of genes might lead to overfitting. I.e., the machine learning algorithm learns structures in the training set which are not representative for the general problem. In order to avoid overfitting, the number of genes should be smaller than half the number of samples in the training set [5]. Thus, gene selection was applied to reduce the number of genes before every GP run.

To this end we developed the GENEACTIVATOR software. This program was used to select relevant genes from the overall data set (with a maximum of 63 genes chosen as inputs of DISCIPULUS). Relevance was measured by different statistical tests in order to build an efficient prediction set with DISCIPULUS. Every procedure calculates a relevance score for all m genes and those genes with the highest scores are selected to set up a new reduced matrix. We tested eight different procedures: Calculating the interval width (IW) of all n values of a gene, the standard deviation (SD) of all n values, the two-partition (2P) criterion, the mean difference (MD) of the components of both classes [28], the signal-to-noise ratio (S2N) [31], the Fisher criterion (FC) [7], the cluster count criterion (CC) and random selection (R) (used as control).

2P and CC had not been taken from literature. For calculating the 2P relevance, we used the following formula:

$$rel_{TWO-PARTITION}(x) = \frac{\mu_{>\mu} - \mu_{\leq\mu}}{\sigma_{>\mu} + \sigma_{\leq\mu}},$$

where $\mu_{>\mu}$ is the mean of the components greater than the mean of all n components and where $\mu_{\leq\mu}$ is the mean of the components less than or equal to the mean of all n components. The standard deviations are named accordingly. The result of 2P describes the ability to divide the n values of a gene into two clusters. For computing the relevance with CC all n values of a gene have to be numerically ordered. Every component belongs to a sample which belongs to one of two classes. After ordering a pointer moves through the array and counts the number of transitions between equal classes. This number is used as the relevance score, where less transitions mean a higher degree of discrimination between the two classes.

Gene selection was always done based on values calculated from the union of training and validation set of the corresponding experiment, and then used for the corresponding applied set. This was necessary, since DISCIPULUS uses the training and the validation set for optimizing the population. GENEACTIVATOR is also able to normalize the data in a discrete or continuous way. In some experiments we used this feature before gene selection. Discrete normalization was used by mapping the expression values to 2,3 or 5 values, for example to 1 and 2. In case of continuous normalization, we scaled the genes to an interval $[1, 100]$.

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

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

2.4 GP runs

After applying selection on the four data sets we used DISCIPULUS to start the GP runs. For most parameters we used the preset preferences of the software which are recommended by the developers (Table 3). These parameters includes a randomization of population size (only restricted by the RAM size of the computer), maximal program size, mutation rate and recombination rate at the assigned values. The results presented here are the best values after a series of 100 internal single GP-runs that are connected on the basis of adaptive termination which is a special feature of DISCIPULUS. That is to say that after a termination of one run, which occurs after 300 generations without improvement in our configuration, a new run is started. If the fitness during runtime is worse than in the run before, the run is discarded and restarted with a certain probability (we used 0.5). One series of these internal runs form one GP experiment and we present the final results in the next section. In other work sample classification with microarrays has been done using n-fold- or leave-one-out cross-validation. This was not feasible in our work, because the DISCIPULUS tool offers no option as of yet. Nevertheless, the danger of overfitting was countered by other measures applied in DISCIPULUS (see section 2.2)

3 Results

In the first experiments we applied all eight gene selection procedures on the two binary data sets. For each of the sixteen reduced data sets experiments were made with the best 10 and 20 genes, respectively. The best hit rate on the applied data set was achieved with 2P (10 genes) on both sets (95% on colon cancer and 100% on ALL/AML) (Table 1 shows one of the 100%-classifiers on the ALL/AML set as an example). That was surprising, since 2P is an unsupervised learning method. On the ALL/AML set, all the supervised methods MD, S2N, FC and CC were also quite successful with an average of 95%. These procedures did not perform equally well on the colon set with hit rates of about 75% to 80%. A reason for this decrease in performance is the higher noise we measured with signal-to-noise calculation in this set. We also used the random selector four times for 5, 10 and 20 genes on each dataset. The results were also surprising, because the hit rates were quite good (up to 91.67%) for a random selection on those large data sets. Obviously there exist many relevant genes in the set, perhaps genes which are only important together with other particular genes.

In the next experiment, we used the same conditions as Golub et al. in their experiments [20]. We used S2N with 50 genes for selection and we achieved a hit rate of 94.12% in team mode. With 2P and 50 genes the hit rate was 100%. We also tested discrete and continuous normalization, but without much success. Only MD improved its hit rate with normalization (with both types), but it still performed worse than other procedures without normalization (90% on colon cancer and 95.83% on ALL/AML). Apparently, it is not advantageous for GP to normalize the data in combination with the examined gene selection methods.

After examination of the small data sets, we analyzed the multiple cancer sets. We started with the 4-class SRBCT set with the 15 best genes of 2P and S2N, because these procedures were the best in the previous runs. For every experiment, we did four runs (OVA), one for every class. Afterwards we used a very simple postprocessing procedure: If the result for a sample was ambiguous or if one classifier made a wrong prediction, we interpreted this as an error. With 2P the hit rate was 32% in single mode and 56% in team mode. This is not surprising because 2P was designed to find genes which are good indicators for one of two classes. With S2N we reached a hit rate of 48% and of 84% in team mode. The samples that were not classified correctly were all classified as unknown. None of the four single classifiers had made a positive prediction. A reason for this behaviour might be the low selection pressure on programs to recognize positive samples, because with an increasing number of classes the number of members per class becomes smaller. In order to compensate this effect we used different class weights. Let Φ be a data set, $\Phi_{class\ i}$ the subset of all samples belonging to class i

Data set	Best GP results of this study	Other EA approaches	Other methods
Colon tumor	95.00% (2P, 10 genes)	100.00% [12]	90.00% [18] 87.09% [1]
ALL/AML	100.00% (2P, 10 genes)	100.00% [12] 97.06% [27]	97.05% [2] 97.05% [10] 97.05% [25] 94.11% [33] 94.11% [18] 85.29% [20]
SRBCT	96.00% (S2N, 15 genes)	100.00% [15]	100.00% [25] 100.00% [33]
GCM	62.96% (S2N, 30 genes)	84.30% [12]	80.00% [3] 78.00% [31]

Table 4: Comparison of the best results on the different applied data sets used in this work. The second column shows our best results (fitness of best single individual). The used method for gene selection and the number of selected genes is presented in brackets.

and s a positive number, for example $s = 100$. Then the weight is calculated by the following formula:

$$Weight_{class\ i}(\Phi) = \frac{s}{2 \cdot |\Phi_{class\ i}|}$$

With these weights, the importance of one class is equal to all other classes in an OVA experiment. In single mode we reached a hit rate of 96%. The hit rate in team mode was very poor with only 68%, because one of the four single classifiers had a very high error rate. We also tested some genes found by Driscoll et al. for being well suited for classification [15]. In their publication a special kind of GP was used to analyze the SRBCT set. They reached a perfect result without any errors with only 10 genes. We took the same genes and built four classifiers (2 with 2 genes and 2 with 3 genes). We used again different class weights. The hit rate was 88% in single mode and 92% in team mode.

In the last experiment we analyzed the 14-class set. Here we developed an algorithm for postprocessing the results of the 14 classifiers by saving the best teams as C-code and putting them together in a small program. Each of these teams deliver a confidence value for its prediction and the number of positive votes in the team. If more than one classifier made a positive prediction, we used the classifier with the highest confidence value for determining the final result (hit rate 57.4%). Alternatively we used the classifier with the highest number of positive votes, leading to better results (62.96%). If no classifier made a positive prediction, we selected the classifier with the lowest confidence or the highest number of positive votes. We also implemented two two-step procedures. If the confidence value of two classifiers is equal, the number of votes is used for predicting the class (59.95%) and the other way round (62.96%). With a simple preprocessing used on the SRBCT set, we just reached a hit rate of 46.29%.

4 Discussion

It has been shown that GP is a suitable method for gene expression data analysis. Table 4 shows a comparison of different papers and projects in which the same data sets had been used. The results for the ALL/AML set is perfect like the outcome of [12] where another EA approach was used. In multi-class classification on more complex data sets the recognition rates are still to be improved compared to other methods, but, nevertheless, are encouraging for further research. Altogether it seems that Evolutionary Algorithms are a useful tool for those problems. It should also be possible to advance the accuracy with Genetic Programming further.

An interesting further step might be a pre-analysis of the data to choose a suitable gene selection method. Perhaps if the signal-to-noise ratio is poor the unsupervised 2-partition method might be preferable. For example, signal-to-noise is disadvantageous for data where the values of one class are

between the values of the other class. Also, it is unclear why the genes found by Driscoll et al. to discriminate well between classes are so important. Only about half of them have a good signal-to-noise value. So the search for better selection procedures is important, particularly for procedures ranking combinations of genes and not only single genes. One gene might be useless alone, but in combination with another certain gene it can be significant for a reliable diagnosis. In this context it would be interesting to apply the methods described in [19] and [26] with our data.

A second useful improvement would be a more complex and specialized GP-system. DISCIPULUS can be applied only to binary classification problems[1]. A better support for non-binary problems is needed, perhaps with automatic OVA or the automatic use of All Pairs (AP). AP is an alternative method to OVA: One class is separated from only one other class. This method is iterated for all pairs of classes with a postprocessing step added [31]. Because of good results with statistical approaches, it might be useful to employ some of those methods for a diagnosis. Weighted voting of different classifiers, calculated in different ways, can be used for reliable results. The weights could be calculated by another machine learning approach. In a final step a support for cross-validation would be useful to get more comparable results.

Acknowledgement

The authors would like to thank Register Machine Technologies Inc, Littleton, CO, for allowing us to use DISCIPULUS (Academic) Version 3.0.

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*, pages 69–81, Boston, USA, 2002. Kluwer Academic Publishers.

[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(14):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. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.

[7] C.-M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, U.K., 1995.

[8] B. Boser, I. Guyon, and V. Vapnik. An training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, 1992.

[9] M.-P.-S. Brown, W.-N. Grundy, D. Lin, and N. et. al Cristianini. Knowledge-based analysis of microarray gene expression data by using support vector machines. *PNAS*, 97(1):262–267, 2000.

[10] S. Busygin, G. Jacobsen, and E. Krdmer. Double conjugated clustering applied to leukemia microarray data. In *2nd SIAM ICDM, Workshop on clustering high dimensional data*, 2002.

[11] J. Cai, A. Dayanik, H. Yu, N. Hasan, T. Terauchi, and W.-N. Grundy. Classification of cancer tissue types by support vector machines using microarray gene expression data. http://citeseer.nj.nec.com/393108.html, 2002.

[12] K. Deb and A.-R. Reedy. Classification of two and multi-class cancer data reliably using multi-objective evolutionary algorithms. Technical Report 2003006, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, June 2003.

[13] K. Deb and A.-R. Reedy. Classification of two-class cancer data reliably using evolutionary algorithms. Technical Report 2003001, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, February 2003.

[1] It works also with regression problems, but the results in projecting a multiple class problem on a regression model were poor (unpublished).

[14] K. Deb and A.-R. Reedy. Reliable classification of two-class cancer data using evolutionary algorithms. *BioSystems*, 72:111–129, 2003.

[15] J.-A. Driscoll, B. Worzel, and D. MacLean. Classification of gene expression data with genetic programming. In Rick L. Riolo, editor, *Genetic Programming: Theory and Practise*. Kluwer, 2003.

[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. Technical report submitted to RECOMB 2000, Hebrew University Jerusalem, 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(10):906–914, 2000.

[19] C. Furlanello, M. Serafini, S. Merle, and G. Jurman. Entropy-based gene ranking without selection bias for the predictive classification of microarray data. *BMC Bioinformatics*, 4(54), 2003.

[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] A.-L. Hsu, S.-L. Tang, and S.-K. Halgamuge. An unsupervised hierarchical dynamic self-organizing approach to cancer class discovery and marker gene identification in microarray data. *Bioinformatics*, 19(16):2131–2140, 2003.

[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 *Papers from CAMDA '00*, 2000.

[24] J. Koza. *Genetic Programming*. MIT Press, Cambridge, MA, 1992.

[25] Y. Lee and C.-K. Lee. Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics*, 19(9):1132–1139, 2003.

[26] J. Lepre, J. Rice, Y. Tu, and G. Stolovitzky. Genes at work: An efficient algorithm for pattern discovery and multivariate feature selection in gene expression data. *Bioinformatics*, 20(7):1033–1044, 2004.

[27] 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.

[28] W. Li and Y. Yang. Zipf's law in importance of genes for cancer classification using microarray data. *J Theor Biol.*, 219(4):539–551, 2002.

[29] D. Michie, D.-J. Spiegelhalter, and C.-C. Taylor. *Machine learning, neural and statistical classification*. Prentice Hall, 1994.

[30] T. Ottmann and P. Widmayer. *Algorithmen und Datenstrukturen*. Spektrum Akademischer Verlag, Heidelberg, 2002.

[31] S. Ramaswamy, P. Tamayo, R. Rifkin, and S. Mukherjee. Multiclass cancer diagnosis using tumor gene expression signatures. *PNAS*, 26(10):15149–15154, 2001.

[32] T. Speed. *Statistical Analysis of Gene Expression Microarray Data*. Chapman and Hall/CRC, Boca Raton, London, New York, Washington D.C., 2003.

[33] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *PNAS*, 99(10):6567–6572, 2001.