Global Top-Scoring Pair Decision Tree for Gene Expression Data Analysis

Marcin Czajkowski and Marek Kretowski

Faculty of Computer Science, Bialystok University of Technology
Wiejska 45a, 15-351 Białystok, Poland
{m.czajkowski,m.kretowski}@pb.edu.pl

Abstract. Extracting knowledge from gene expression data is still a major challenge. Relative expression algorithms use the ordering relationships for a small collection of genes and are successfully applied for micro-array classification. However, searching for all possible subsets of genes requires a significant number of calculations, assumptions and limitations. In this paper we propose an evolutionary algorithm for global induction of top-scoring pair decision trees. We have designed several specialized genetic operators that search for the best tree structure and the splits in internal nodes which involve pairwise comparisons of the gene expression values. Preliminary validation performed on real-life micro-array datasets is promising as the proposed solution is highly competitive to other relative expression algorithms and allows exploring much larger solution space.

Keywords: evolutionary algorithms, decision tree, top-scoring pair, classification, gene expression, micro-array.

1 Introduction

DNA chips [16] may be used to assist diagnosis and to discriminate cancer samples from normal ones [17]. Extracting accurate and simple decision rules that contains marker genes are of great interest for biomedical applications. However, finding a meaningful and robust classification rule is a real challenge, since in different studies of the same cancer, diverse genes consider to be marked [23].

Dimensionality and redundancy are one of the most typical statistical problems that often occur with micro-array analysis. In particular, we are faced with the *"small N, large P problem"* [27] of statistical learning. The number of samples (denoted by N) comparing to the number of genes (P) remains quite small as N usually does not exceeded one or two hundreds where P is usually several thousands. The high ratio of features/observations may influence the model complexity and can cause the classifier to over-fit the training data. Furthermore, most of genes are known to be irrelevant so the gene selection prior to classification should be considered [17] to: simplify calculations, decrease model complexity and often to improve accuracy of the following classification.

Recently, a large number of supervised solutions have been described in literature for micro-array classification, including: nearest neighbors [8], neural

K. Krawiec et al. (Eds.): EuroGP 2013, LNCS 7831, pp. 229–240, 2013.

networks [3], Support Vector Machine [20] and random forests [7]. Most of machine learning methods provide "black box" decision rules, which usually involve many genes combined in a highly complex fashion and therefore are difficult to interpret from medical point of view. There is a need for simple models like decision trees or rule extraction systems which may actually help in understanding and identifying casual relationships between specific genes.

In this paper we propose a hybrid solution called Global Top-Scoring Pair Decision Tree ($GTSPDT$) that combines the power of evolutionary approach, relative expression algorithms and decision trees. It combines different top-scoring extensions, eliminates their restrictions and allows exploring much larger solution space. Evolutionary algorithm (EA) globally searches for the best tree structure and tests which involve pairwise comparisons of the gene expression values. The general structure of our solution follows a typical framework of EA with an unstructured population and a generational selection. We have designed several specialized operators to mutate and cross-over individuals and a fitness function that helps mitigating the over-fitting problem.

The rest of the paper is organized as follows. In the next section the relative expression algorithms and decision tree classifiers for gene expression analysis are briefly recalled. Section 3 describes in detail the $GTSPDT$ solution and section 4 presents preliminary experimental validation on real-life micro-array datasets. In the last section, the paper is concluded and possible future works are sketched.

2 Background and Motivation

In this section the decision trees and the family of top-scoring algorithms are presented and their application for gene expression data is discussed.

2.1 Decision Trees

Decision trees (also known as classification trees) [22] represent one of the main techniques of classification analysis in data mining and knowledge discovery. They predict the class membership (dependent variable) of an instance using its measurements of predictor variables.

In the literature, there are several attempts to use decision trees for the classification analysis on gene expression data. In [8] the author compares some classification principles, among which there is the CART system and in [28] the application of C4.5, bagged and boosted decision trees are presented. In [32] the author compares decision trees with SVMs on gene expression data and concludes that bagging and boosting decision tress perform as well as or close to SVM algorithms. However ensemble methods and decision trees with complex multivariate tests based on linear or non-linear combination splits are much more difficult to understand or interpret by human experts. Although higher accuracy than single-tree solutions, their potential for scientific modeling of underlying processes is limited.

2.2 A Family of Top-Scoring Algorithms

Relative expression algorithms [10] are simple yet powerful classifiers. The use of the ordering relationships for a small collection of genes has potential for identify gene-gene interactions with plausible biological interpretation and direct clinical applicability [15]. The most popular solution is called Top-Scoring Pair (TSP) [10] and has many applications in identifying marker genes in micro-array datasets [26] or as a feature selection in more complex classifiers [32]. In addition, the TSP solution is parameter free, data driven learning approach that is invariant to any simple transformation of data like normalization and standardization.

TSP is extended in two main directions, each having its pros and cons. First technique called $k-TSP$ [29] increases the number of top-scoring pairs included in the final prediction. This solution was later extended by weight pairwise comparisons $Weight\ k-TSP$ [4] and Top-Scoring Pair Decision Tree ($TSPDT$) [5]. Different approaches called Top-Scoring Triplet (TST) [15] and Top-Scoring 'N' (TSN) [19] search for more than two ordering relationships between genes. Multiple implementation of these solutions may be found as R package [31].

Top-Scoring Pair. The TSP method proposed by Donald Geman [10] is based on pairwise comparisons of gene expression values. Discrimination between two classes depends on finding pairs of genes that achieve the highest ranking value called "score". Consider a gene expression profile consisting of P genes and N samples participating in the training micro-array dataset. Let the data be represented as a $P \times N$ matrix in which expression value of u-th gene from v-th sample is denoted as x_{uv}. Each row represents observations of a particular gene over N training samples, and each column represents a gene expression profile composed from P genes. Each profile has a true class label denoted $C_m \in C = \{C_1, \ldots, C_M\}$. For the simplicity of calculations it is assumed that there are only two classes ($M = 2$) and profiles with indexes from 1 to N_1 ($N_1 < N$) belong to the first class (C_1) and profiles from range $\langle N_1 + 1, N\rangle$ to the second class (C_2).

The TSP method focuses on gene pair matching (i, j) $(i, j \in \{1, \ldots, P\}, i \neq j)$ for which there is the highest difference in probability p of an event $x_{in} < x_{jn}$ $(n = 1, 2, \ldots, N)$ between class C_1 and C_2. For each pair of genes (i, j) two probabilities are calculated $p_{ij}(C_1)$ and $p_{ij}(C_2)$:

$$p_{ij}(C_1) = \frac{1}{|C_1|} \sum_{n=1}^{N_1} I(x_{in} < x_{jn}),$$

$$p_{ij}(C_2) = \frac{1}{|C_2|} \sum_{n=N_1+1}^{N} I(x_{in} < x_{jn}),$$

where $|C_m|$ denotes a number of profiles from class C_m and $I(x_{in} < x_{jn})$ is the indicator function defined as:

$$I(x_{in} < x_{jn}) = \begin{cases} 1, & \text{if } x_{in} < x_{jn} \\ 0, & \text{if } x_{in} \geq x_{jn} \end{cases}$$

TSP is a rank-based method, so for each pair of genes (i, j) the "score" denoted Δ_{ij} is calculated as:

$$\Delta_{ij} = |p_{ij}(C_1) - p_{ij}(C_2)|.$$

In the next step of the algorithm, pairs with the highest score are chosen.

There should be only one top pair in the TSP method, however it is possible that multiple gene pairs achieve the same top score. In that case a secondary ranking, based on the rank differences in each class and samples, is used to eliminate draws.

$$\gamma_{ij}(C_1) = \frac{\sum_{n=1}^{N_1}(x_{in} - x_{jn})}{|C_1|},$$

$$\gamma_{ij}(C_2) = \frac{\sum_{n=N_1+1}^{N}(x_{in} - x_{jn})}{|C_2|}.$$

For each pair of genes (i, j) the second ranking is calculated and pair with the highest score τ_{ij} is chosen:

$$\tau_{ij} = |\gamma_{ij}(C_1) - \gamma_{ij}(C_2)|,$$

The TSP prediction is made by comparing the relation between expression values of two genes (i, j) marked as "top-scoring pair" in new test sample w. If we observe that $p_{ij}(C_1) \geq p_{ij}(C_2)$ and $x_{iw} < x_{jw}$, then TSP votes for class C_1, however if $x_{iw} \geq x_{jw}$ then TSP votes for class C_2. An opposite situation is when $p_{ij}(C_1) < p_{ij}(C_2)$, cause if $x_{iw} < x_{jw}$ TSP votes for C_1 and if $x_{iw} \geq x_{jw}$ TSP chooses C_2.

Top-Scoring Extensions. There are two main ways to extend the TSP solution: application of multiple pairs of genes or comparison relationships for more than two genes. One of the solutions that uses the first approach is $k-TSP$ [29] which applies no more than k top-scoring pairs in classification. The parameter k can be set up a priori or can be determined by a cross-validation. Next, the $k-TSP$ classifier uses no more than k top scoring disjoint gene pairs that have the highest score and simple majority vote for a final decision.

The $Weight\ k-TSP$ [4] solution modifies rankings of $k-TSP$ and calculates the ratio of two genes in order to find optimal top-scoring pairs.

Solution called $TSPDT$ [5] is a hybrid of $k-TSP$ and a top-down induced decision tree [24]. At first, a test analogous to the $k-TSP$ method is searched for the root node. Then, the set of instances is split according to decision of the best pair (or pairs) of genes in the current node and then each derived subset goes to the corresponding branch. The process is recursively repeated for each branch until leaf node is reached.

Different approach for the TSP extension is discussed in [15] where authors focused on the predicting germline BRCA1 mutations in breast cancer. A three-gene version of relative expression analysis called Top-Scoring Triplet (TST) [15] was proposed as potentially more discriminating than TSP since there are six possible orderings that must be analyzed.

Next, the general idea of pairwise or triplet rank comparisons was proposed in [19]. The top-scoring N (TSN) algorithm uses generic permutations and dynamically adjust the size to control both the permutation and combination space available for classification. Variable N denotes the size of the classifier, therefore in the case where $N = 2$ the TSN algorithm simply reduces to the TSP method and when $N = 3$, the TSN can be seen as TST. The classifier's size can be chosen by a user or by an internal cross-validation that checks classification accuracy for the different values of N (on a training data, in a range specified by the user) and selects the classifier with the highest score.

2.3 Motivation

There are two main drawbacks of TSP extensions. The first one is enormous computational requirements because the general complexity of aforementioned algorithms is $O(k * P^N)$, where k is the number of top-scoring groups, P is the number of features and N is the size of group of genes which ordering relationships is compared. There are some attempts of improving TSP performance by parallelization the algorithm and using graphic processing unit (GPU) for calculations [18], however the parameters k or/and N must be small (upper limit of the test was equal: $N = 4$, $k = 1$ but only when P was significantly reduced by the feature selection).

The second drawback is finding accurate value of the parameters k and N. In TSP extensions they are defined by the user or determined by internal cross-validation. However, it is time consuming and decreases the set of instances which is already very small. In addition, it is also not clear which extension should be prefered: $k - TSP$ or TSN. It should be noted that the $k - TSP$ algorithms cannot replace the TSN with $N > 2$ as the $k - TSP$ has restriction to use only disjoint gene pairs. On the other side, the $k - TST$ or $k - TSN$ were not proposed in the literature, probably because of it's huge complexity.

In the $TSPDT$ system $k - TSP$ algorithm is calculated in each non-terminal tree node, therefore the general complexity must be multiplied by the number of internal nodes. In addition, the $TSPDT$ like most of practical decision-tree inducers is based on heuristics such as greedy approach where locally optimal decisions are made in each node and cannot guarantee to return optimal classifier.

Previously performed research showed that decision trees [11,6], extension of TSP [4] and hybrid solution called $TSPDT$ [5] may be successfully applied to the gene expression data. In this paper we would like to unite aforementioned extensions of TSP through the evolutionary approach. We propose a hybrid solution called Global Top-Scoring Pair Decision Tree ($GTSPDT$) that combines the power of evolutionary approach, relative expression algorithms and decision trees.

Our goal is to improve classification accuracy and help in identifying genomic "marker interactions". Evolutionary algorithm searches for the best tree structure and tests which involve multiple pairwise comparisons of the gene expression values. The number of top-scoring pairs applied in each split is determined by the evolution and by removing restrictions on disjoint gene pairs, the splits may

compare relationships for more than two genes like in TSN. Application of evolutionary algorithms to the TSP solutions can decrease computation time and allows to explore larger solution space.

3 Global Top-Scoring Pair Decision Tree

General structure of $GTSPDT$ follows a typical framework of evolutionary algorithms [21] with an unstructured population and a generational selection.

Representation. Decision trees are quite complicated tree structures, in which number of nodes, type of the tests and even number of test outcomes are not known in advance. Therefore, representing individuals in their actual form (as potential tree-solutions) seems more adequate than encoding them in the fixed-size (usually binary) chromosomes.

Figure 1 illustrates the single individual. Each test in a non-terminal node is composed of a group of top-scoring pairs. Similarly to $TSPDT$ and $k-TSP$, the final decision in each node depend on a simple majority voting where each top-scoring pair vote has the same weight. Therefore, the TST solution can be represented by the 3 top-scoring pairs that involve only three genes. In the analogous way, TSN, $k-TSP$ or even a variation $k-TSN$ representation can be found by the $GTSPDT$. In every node information about learning vectors associated with the node is also stored. This enables the algorithm to perform more efficiently local structure and tests modifications during applications of genetic operators.

Initialization. Initial population could be generated randomly to cover the entire range of possible solutions, however due to the large solution space, seeding the initial population with good solutions may speed up evolutionary search. Each individual in the initial population is generated by the classical top-down,

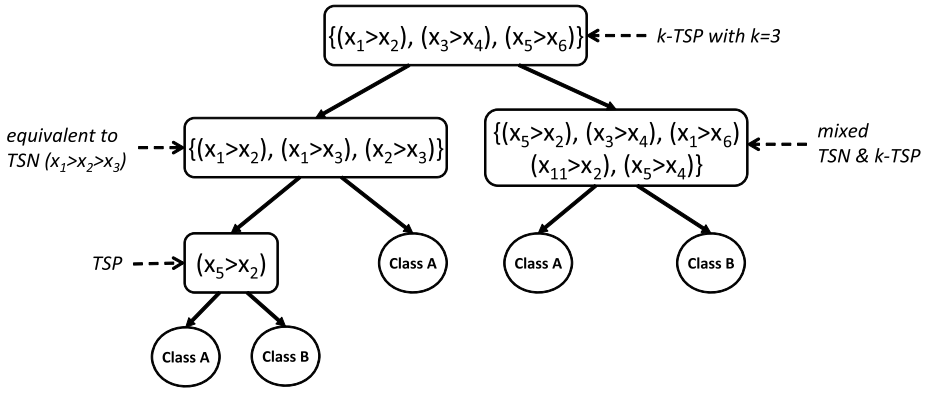

Fig. 1. An example representation of a single individual with different tests in internal nodes

greedy approach. Split in each internal node is based on a mixed dipole strategy [13] and constructed as follows. Among feature vectors located in the node two objects from different classes are randomly chosen. Next, an effective top-scoring pair test (one pair of genes which separates this two objects) constructed on randomly selected attributes constitute a split. The recursive partitioning is finished when the node is pure (all training objects in the node are from the same class) or the number of objects is lower than the predefined value (default value: 5).

Selection and Termination Condition. Ranking linear selection [21] is applied as a selection mechanism. In each iteration, single individual with the highest value of fitness function in current population is copied to the next one *(elitist strategy)*. Evolution terminates when the fitness of the best individual in the population does not improve during the fixed number of generations (default value: 1000). In case of a slow convergence, maximum number of generations is also specified (default value: 10000), which allows to limit the computation time.

Genetic Operators. To maintain genetic diversity, we have proposed two specialized genetic operators corresponding to the classical mutation and cross-over. Each evolutionary iteration starts with randomly choosing the operator type where the default probability to select mutation equals 0.8 and to select cross-over equals 0.2. Both operators have impact on the tree structure and the tests in non-terminal nodes. After each operation it is usually necessary to relocate learning vectors between parts of the tree rooted in the altered node. This can cause that certain parts of the tree does not contain any learning vectors and has to be pruned.

Cross-over starts with selecting positions in two affected individuals. We have adapted three variants of recombination [13]:

- subtrees starting in the selected nodes are exchanged;
- tests associated with the nodes are exchanged (only when non-terminal nodes are chosen);
- branches which start from the selected nodes are exchanged in random order (only when non-terminal nodes are chosen).

Mutation solution starts with randomly choosing the type of node (equal probability to select leaf or internal node). Next, the ranked list of nodes of the selected type is created and a mechanism analogous to ranking linear selection is applied to decide which node will be affected. Depending on the type of node, ranking takes into account two elements:

- location (level) of node. It is evident that modification of the test in the root node affects whole tree and has a great impact, whereas mutation of an internal node in lower parts of the tree has only a local impact. Therefore, internal nodes in lower parts of the tree are mutated with higher probability;
- classification accuracy of the node - worse in terms of prediction accuracy leaves and internal nodes are mutated with higher probability (homogeneous leaves are not included).

Each leaf can be transformed into an internal node with a new dipole test, similar to one used in population initialization. As for the internal nodes, we have propose a few variants of mutation:

- node can be transformed (pruned) into a leaf,
- test in node is replaced by new top-scoring pair,
- one of the attributes from top-scoring pair is replaced by random one which effectively separates at least two objects in the node,
- new top-scoring pair is added or removed from the test in the node,
- tests between father and son exchanged,
- all subtrees are replaced with randomly chosen one.

Fitness Function. Specification of a suitable fitness function is one of the most important and sensitive element in the design of evolutionary algorithm. It drives the evolutionary search process and measures how good a single individual is in terms of meeting the problem objective. Direct minimization of the prediction error measured on the learning set usually leads to the over-fitting problem. In typical top-down tree inducers it is partially mitigated by a stopping condition and an application of the post-pruning [9].

In case of evolutionary induced classification trees, we need to balance the reclassification quality and the complexity of the tree. A similar idea is used in cost complexity pruning in the CART system [2]. The fitness function is maximized and has the following form:

$$Fitness(T) = Q_{Reclass}(T) - \alpha \cdot (2 * S(T) + K(T)),$$

where $Q_{Reclass}(T)$ is the reclassification quality of the tree T, $S(T)$ is the size of the tree expressed as a number of nodes, K is the number of unique genes that were used to build the classifier and α is the relative importance of the complexity term specified by user (default value is 0.05). Penalty associated with the classifier complexity increases proportionally with the tree size and the number of different genes that constitute the top-pairs to prevent over-fitting.

It should be noticed that there is no optimal value of α for all possible datasets and tuning it may lead to the improvement of results for the specific problem. Further research to determine the appropriate value of complexity penalty term for proposed solution is required and other commonly used measures such as Akaikes information criterion (AIC) [1] or Bayesian information criterion (BIC) [25]should be considered.

4 Results and Discussions

Performance of classifiers was investigated on public available micro-array datasets, summarized in Table 1. We have extend previous comparison of TSP-family algorithms [5] by enclosing the accuracy and the size of proposed solution $GTSPDT$. To check and compare results of other popular decision trees and rule classifiers on analyzed data please also refer to [5].

Table 1. Details of Kent Ridge Bio-medical gene expression datasets

Datasets	Symbol	Attributes	Train	Test
Breast Cancer	BC	24481	34/44	12/7
Central Nervous System	CNS	7129	21/39	-
Colon Tumor	CT	6500	40/22	-
DLBCL vs Follicular Lymphoma	DF	6817	58/19	-
Leukemia ALL vs AML	LA	7129	27/11	20/14
Lung Cancer Brigham	LCB	12533	16/16	15/134
Lung Cancer University of Michigan	LCM	7129	86/10	-
Lung Cancer - Totonto, Ontario	LCT	2880	24/15	-
Ovarian Cancer	OC	15154	91/162	-
Prostate Cancer	PC	12600	52/50	27/8

Datasets and Setup. Proposed solution was tested on Kent Ridge Bio-medical Repository [12] and the datasets refer to the studies of human cancer, including: leukemia, colon tumor, prostate cancer, lung cancer, breast cancer, ovarian cancer etc. If datasets, described in Table 1 were not pre-divided into the training and the testing sets we use typical 10-fold cross-validation. To ensure stable results, for all datasets average score of 10 runs is shown.

In the experiments, we have compared proposed solution with TSP, $k-TSP$ and $TSPDT$. The maximum number of top-scoring pairs (parameter k) for $k-TSP$ and $TSPDT$ was set to 9. Classification was performed with default parameters for all algorithms through all datasets and was preceded by a step known as feature selection, where a subset of relevant features is identified. We decided to use popular method called Relief-F [14] for micro-array analysis with its default parameters and 1000 features subset size.

Comparison of Top-Scoring Family Algorithms Methods. Table 2 summaries classification performance for the proposed solution TSP, $k-TSP$, $TSPDT$ and $GTSPDT$. Preliminary results show that on most of datasets, the classification accuracy increased (or did not change) when decision trees with TSP were applied. However, for some datasets, like Colon Tumor, both decision tree solutions did not work well which may suggest over-fitting to the training data. In general $GTSPDT$ managed to increase classification accuracy (average on all datasets over 3%). The greatest improvement of $GTSPDT$ can be noticed on the Lung Cancer datasets. According to the Friedman test, there is a statistically significant difference (p-value of 0.0019) in the accuracy between TSP and $GTSPDT$.

Number of internal nodes and the average number of top-scoring pairs used in $GTSPDT$ classifier presented in Table 2 allows to compare the sizes of tested solutions. The TSP algorithm uses only one pair of genes and $k-TSP$ no more than 9 pairs. The $TSPDT$ tree uses no more than $k=9$ pairs in each internal node, so this value must be multiplied by the tree size. The proposed solution managed to slightly decrease the tree size comparing to $TSPDT$ and used less pairs of genes in each internal node (an average: 2.2).

Table 2. Comparison of top-scoring algorithms, including accuracy, number of internal nodes and the number of gene pairs

Datasets	**Classifiers accuracy and size of the solution**						
	TSP	k-TSP	TSPDT		GTSPDT		
	accuracy	accuracy	nodes	accuracy	nodes	pairs	accuracy
BC	52.63	68.42	2.0	78.95	1.1	2.9	77.37
CNS	49.00	58.50	3.0	63.00	1.1	3.1	65.00
CT	83.64	88.93	2.0	84.88	1.8	2.6	82.26
DF	72.75	87.82	1.6	95.25	1.4	3.2	97.70
LA	73.53	91.18	1.0	91.18	1.0	1.0	91.18
LCB	76.51	83.89	1.0	83.89	1.0	2.5	93.02
LCM	95.87	95.23	1.1	97.77	1.0	1.1	98.96
LCT	50.92	58.42	2.4	55.33	1.6	2.7	78.46
OC	99.77	100.00	1.0	100.00	1.0	1.0	99.60
PC	76.47	91.18	2.0	94.12	2.2	1.9	91.76
Average	73.11	82.36	1.7	84.44	1.3	2.2	87.53

5 Conclusion

In this paper we propose the *GTSPDT* system for solving classification problems on micro-array data. The evolutionary approach of the hybrid solution combines the power of decision trees and popular top-scoring algorithms. EA globally searches for the best tree structure and the top-scoring pairs which are used as splitting tests in non-terminal nodes. We have designed several specialized operators to mutate and cross-over individuals (trees) and a fitness function that helps mitigating the over-fitting problem. The *GTSPDT* solution is highly competitive to other relative expression algorithms in terms of accuracy and the model complexity. It can explore much larger permutation and combination space and therefore has potential to discover new biological connections between genes.

In this paper we only focus on the general concept of *GTSPDT*. We do not enclose any biological aspects of the rules generated by proposed system or case studies on particular datasets. Furthermore improvement is still required. Application of local optimizations (memetic algorithms), new specialized operators and self-adaptive parameters should speed up convergence of the evolutionary algorithm. We also want to test different fitness functions based on e.g. information criterion and extended *GTSPDT* to handle cost-sensitive and multi-class problems. More work on preprocessing datasets, gene selection and using additional problem-specific knowledge is also required to improve *GTSPDT* classification and rule discovery.

Acknowledgements. This work was supported by the grant S/WI/2/13 from Bialystok University of Technology.

References

1. Akaike, H.: A New Look at Statistical Model Identification. IEEE Transactions on Automatic Control 19, 716–723 (1974)
2. Breiman, L., Friedman, J.: Classification and Regression Trees. Wadsworth Int. Group (1984)
3. Cho, H.S., Kim, T.S.: cDNA Microarray Data Based Classification of Cancers Using Neural Networks and Genetic Algorithms. Nanotech 1 (2003)
4. Czajkowski, M., Kretowski, M.: Novel Extension of $k - TSP$ Algorithm for Microarray Classification. In: Nguyen, N.T., Borzemski, L., Grzech, A., Ali, M. (eds.) IEA/AIE 2008. LNCS (LNAI), vol. 5027, pp. 456–465. Springer, Heidelberg (2008)
5. Czajkowski, M., Kretowski, M.: Top Scoring Pair Decision Tree for Gene Expression Data Analysis. In: Software Tools and Algorithms for Biological Systems. Advances in Experimental Medicine and Biology, vol. 696, pp. 27–35 (2011)
6. Czajkowski, M., Grześ, M., Kretowski, M.: Multi-Test Decision Trees for Gene Expression Data Analysis. In: Bouvry, P., Kłopotek, M.A., Leprévost, F., Marciniak, M., Mykowiecka, A., Rybiński, H. (eds.) SIIS 2011. LNCS, vol. 7053, pp. 154–167. Springer, Heidelberg (2012)
7. Diaz-Uriarte, R., Alvarez de Andres, S.: Gene selection and classification of microarray data using random forest. BMC Bioinformatics 7, 3 (2006)
8. Dudoit, S.J., Fridlyand, J.: Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association 97, 77–87 (2002)
9. Esposito, F., Malerba, D., Semeraro, G.: A comparative analysis of methods for pruning decision trees. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(5), 476–491 (1997)
10. Geman, D., d'Avignon, C., Naiman, D.Q., Winslow, R.L.: Classifying gene expression profiles from pairwise mRNA comparisons. Statistical Applications in Genetics and Molecular Biology 3(19) (2004)
11. Grześ, M., Kretowski, M.: Decision Tree Approach to Microarray Data Analysis. Biocybernetics and Biomedical Engineering 27(3), 29–42 (2007)
12. Kent Ridge Bio-medical Dataset Repository, `http://datam.i2r.a-star.edu.sg/datasets/index.html`
13. Kretowski, M., Grześ, M.: Evolutionary Induction of Mixed Decision Trees. International Journal of Data Warehousing and Mining 3(4), 68–82 (2007)
14. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
15. Lin, X., Afsari, B., Marchionni, L., Cope, L., Parmigiani, G., Naiman, D., Geman, D.: The ordering of expression among a few genes can provide simple cancer biomarkers and signal BRCA1 mutations. BMC Bioinformatics 10(256) (2009)
16. Lockhart, D.J., Winzeler, E.A.: Genomics, gene expression and DNA arrays. Nature 405, 827–836 (2000)
17. Lu, Y., Han, J.: Cancer classification using gene expression data. Information Systems 28(4), 243–268 (2003)
18. Magis, A.T., Earls, J.C., Ko, Y., Eddy, J.A., Price, N.D.: Graphics processing unit implementations of relative expression analysis algorithms enable dramatic computational speedup. Bioinformatics 27(6), 872–873 (2011)
19. Magis, A.T., Price, N.D.: The top-scoring 'N' algorithm: a generalized relative expression classification method from small numbers of biomolecules. BMC Bioinformatics 13(1), 227 (2012)

20. Mao, Y., Zhou, X.: Multiclass Cancer Classification by Using Fuzzy Support Vector Machine and Binary Decision Tree With Gene Selection. Journal of Biomedicine and Biotechnology, 160–171 (2005)
21. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3rd edn. Springer (1996)
22. Murthy, S.: Automatic construction of decision trees from data: A multi-disciplinary survey. Data Mining and Knowledge Discovery 2, 345–389 (1998)
23. Nelson, P.S.: Predicting prostate cancer behavior using transcript profiles. Journal of Urology 172, 28–32 (2004)
24. Rokach, L., Maimon, O.: Top-down induction of decision trees classifiers - A survey. IEEE Transactions on Systems, Man, and Cybernetics - Part C 35(4), 476–487 (2005)
25. Schwarz, G.: Estimating the Dimension of a Model. The Annals of Statistics 6, 461–464 (1978)
26. Shi, P., Ray, S., Zhu, Q., Kon, M.A.: Top scoring pairs for feature selection in machine learning and applications to cancer outcome prediction. BMC Bioinformatics 12(375) (2011)
27. Simon, R., Radmacher, M.D.: Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. Journal of the National Cancer Institute 95, 14–18 (2003)
28. Tan, A.C., Gilbert, D.: Ensemble machine learning on gene expression data for cancer classification. Applied Bioinformatics 2, 75–83 (2003)
29. Tan, A.C., Naiman, D.Q.: Simple decision rules for classifying human cancers from gene expression profiles. Bioinformatics 21, 3896–3904 (2005)
30. Quinlan, R.: Inductive knowledge acquisition: A case study, vol. 9, pp. 157–173. Addison-Wesley (1987)
31. Yang, X., Liu, H.: Top Scoring Pair based methods for classification (BigTSP R package) (2012), `http://cran.r-project.org`
32. Yoon, S., Kim, S.: k-Top Scoring Pair Algorithm for feature selection in SVM with applications to microarray data classification. Soft Computing - A Fusion of Foundations, Methodologies and Applications, 151–159 (2009)