Generic Relative Relations in Hierarchical Gene Expression Data Classification

Marcin Czajkowski(✉), Krzysztof Jurczuk, and Marek Kretowski

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

Abstract. Relative Expression Analysis (RXA) plays an important role in biomarker discovery and disease prediction from gene expression profiles. It deliberately ignores raw data values and investigates only the relative ordering relationships between a small group of genes. The classifiers constituted on that concept are therefore robust to small data perturbations and normalization procedures, but above all, they are easy to interpret and analyze.

In this paper, we propose a novel globally induced decision tree in which node splits are based on the RXA methodology. We have extended a simple ordering with a more generic concept that also explores fractional relative relations between the genes. To face up to the newly arisen computational complexity, we have replaced the typical brute force approach with an evolutionary algorithm. As this was not enough, we boosted our solution with the OpenMP parallelization, local search components calculated on the GPU and embedded ranking of genes to improve the evolutionary convergence. This way we managed to explore in a reasonable time a much larger solution space and search for more complex but still comprehensible gene-gene interactions. An empirical investigation carried out on 8 cancer-related datasets shows the potential of the proposed algorithm not only in the context of accuracy improvement but also in finding biologically meaningful patterns.

Keywords: Evolutionary data mining · Relative Expression Analysis · Decision trees · Gene Expression Data

1 Introduction

Data mining is an umbrella term covering a broad range of tools and techniques for extracting hidden knowledge from large quantities of data. Biomedical data can be very challenging due to the enormous dimensionality, biological and experimental noise as well as other perturbations. Unfortunately, many traditional machine learning algorithm use complex predictive models, which impede biological understanding and are an obstacle for mature applications [1]. Most of the research effort tends to focus almost exclusively on the prediction accuracy of core data mining tasks (e.g., classification and regression), and far less

T. Bäck et al. (Eds.): PPSN 2020, LNCS 12270, pp. 372–384, 2020.
https://doi.org/10.1007/978-3-030-58115-2_26

effort has gone into understand and interpret the discovered knowledge. It is not enough to simply produce good outcomes but to provide logical reasoning just as clinicians do for medical treatments.

There is a strong need for 'white box' computational methods to effectively and efficiently carry out the predictions using biomedical data. One of the example approaches which may actually help in understanding and identifying relationships between specific features and improve biomarker discovery is the Relative Expression Analysis (RXA) [9]. It is a powerful collection of easily interpretable algorithms that plays an important role in genomic data classification [11]. RXA's key novelty is the use of interactions between a small collection of genes by examining the relative order of their expressions rather than their raw values. The influence of RXA solutions could be even greater, however, the simplicity of model decisions which is based only on the plain ordering comparisons strongly limits the search for other gene-gene relations. Additionally, a typical exhaustive search performed by most of RXA solutions limits the number of genes that can be analyzed [16] due to computational complexity.

In this paper, we introduce a new approach for RXA called Evolutionary Relative Expression Decision Tree (Evo-REDT). We have extended the simple ordering relations between the genes proposed in RXA with a new more generic concept. It explores relative fraction comparison in the gene pairs, therefore, it can identify percent changes in their relations between different expression profiles. To include also the hierarchical relations between the gene pairs, we have adapted an evolutionary induced decision tree system called Global Decision Tree (GDT) [15]. It allows performing a simultaneous search for the tests in the internal nodes as well as the overall tree structure. In each splitting node of a tree, we use a test consisting of two genes and a fraction which represents the ratio (weight) of their relations. Originally, the selection of a top pair in RXA performs an exhaustive search for all possible order relations between two genes. Using brute force within the proposed approach is computationally infeasible, on the other hand, relying only on the evolutionary search may result in a very slow algorithm convergence. Therefore, we have proposed several improvements in order to boost our solution, mainly:

- several specialized variants of mutation and crossover operators;
- local search components calculated on the GPU;
- embedded ranking of genes in order to consider the relations based on top genes more often;
- parallel processing of the individuals of the population using shared memory (OpenMP) paradigm.

Our main objective is to find in a reasonable time more advanced relations in comparison to RXA that are more accurate and still easy to understand and interpret.

2 Background

Genomic data is still challenging for computational tools and mathematical modeling due to the high ratio of features to observations as well as enormous gene redundancy and ubiquitous noise. Nearly all off-the-shelf techniques applied to genomics data [1], such as neural networks, random forests and SVMs are 'black box' solutions which often involve nonlinear functions of hundreds or thousands of genes and complex prediction models. Currently, deep learning approaches have been getting attention as they can better recognize complex features through representation learning with multiple layers. However, we know very little about how such results are derived internally. In this section, we focus on two concepts which are the main elements of the proposed approach.

2.1 RXA Classification Algorithms

Relative Expression Analysis focuses on finding interactions among a small group of genes and studies the relative ordering of their expression values. In the pioneer research [10], authors used ranks of genes instead of their raw values and introduced the Top Scoring Pair (TSP) classifier. It is a straightforward prediction rule that makes a pairwise comparison of gene expression values and searches for a single pair of genes with the highest rank. Let x_i and x_j $(0 \leq i, j < N)$ be the expression values of two different genes from available set of genes and there are only two classes: *normal* and *cancer*. First, the algorithm calculates the probability of the relation $x_i < x_j$ between those two genes in the objects from the same class:

$$P_{ij}(normal) = Prob(x_i < x_j | Y = normal) \tag{1}$$

and

$$P_{ij}(cancer) = Prob(x_i < x_j | Y = cancer), \tag{2}$$

where Y denotes the class of the objects. Next, the score for this pair of genes (x_i, x_j) is calculated:

$$\Delta_{ij} = |P_{ij}(normal) - P_{ij}(cancer)|. \tag{3}$$

This procedure is repeated for all distinct pairs of genes and the pair with the highest score becomes the top-scoring pair. In the case of a draw, a secondary ranking that relies on gene expression differences is used [19]. Finally, for a new test sample, the relation between expression values of the top pair of genes is checked. If the relation holds, then the TSP predictor votes for the class that has higher probability P_{ij} in the training set, otherwise it votes for the class with smaller probability.

There are many extensions of the TSP classifier. The main ones focused on increasing the number of gene pairs in the predictive model (k-TSP [19]) or analyzing the order of relationships for more than two genes (TSN [16]). Those methods were also combined with a typical decision tree algorithm (TSPDT [3])

in which each non-terminal node of the tree divides instances according to a splitting rule that is based on TSP or k-TSP accuracy. As one of the main drawbacks of the aforementioned solutions was the enormous computational complexity resulting from the exhaustive search, various optimization techniques were proposed. Some of them were based on parallel computing using GPGPU [16], others used the heuristic approach involving evolutionary algorithms (EA) like EvoTSP [4]. Finally, there are many variations of ranking and grouping the gene pairs [9,13] but all the systems inherited the standard RXA methodology based on the ordering relations.

2.2 Decision Trees

Decision trees have a knowledge representation structure made up of nodes and branches, where: each internal node is associated with a test on one or more attributes; each branch represents the test outcome, and each leaf (terminal node) is designed by a class label. Induction of optimal DT for a given dataset is a known NP-complete problem. As a consequence, practical DT learning algorithms must be heuristically enhanced. The most popular type of tree induction is based on a top-down greedy search [14]. It starts from the root node, where the locally optimal split (test) is searched according to the given optimality measure. Next, the training instances are redirected to the newly created nodes, and this process is repeated for each node until a stopping condition is met. Inducing the DT through a greedy strategy is fast and generally efficient in many practical problems, but it usually produces overgrown solutions.

Evolutionary induction of decision trees is an alternative to greedy top-down approaches as it mitigates some of the negative effects of locally optimal decisions [15]. The strength of such an approach lies in a global search for the tree structure and the tests in the internal nodes. This global induction is much more computationally complex; however, it can reveal hidden regularities that are often undetectable by greedy methods. Unfortunately, there are not so many new solutions in the literature that focus on the classification of genomic data with comprehensive DT models. In the liteature, there is far more interest in trees as sub-learners of an ensemble learning approach, such as Random Forests. These solutions alleviate the problem of low accuracy by averaging or adaptive merging of multiple trees. However, when modeling is aimed at understanding basic processes, such methods are not so useful due to the complexity of the generated rules.

2.3 Motivation

RXA solutions deliberately replace the raw expression data values with simple ordering relationships between the features. However, in a nutshell, limiting knowledge to the information that expression of one gene x_i is larger than another x_2 which has a form of a pair: $(x_i > x_j)$ may result in a large loss of potentially important data. We propose an additional fractional component

called relational weight w, which is the ratio of the genes relation in a pair: $(x_i > w * x_j)$.

Let us hypothetically assume that the two genes x_1 and x_2 have constant expression values among the instances from the same classes. Figure 1 shows three simple scenarios (a), (b), (c) of possible relations between genes x_1 and x_2 in a normal and cancer class. The RXA algorithms will detect only the pairs (x_1, x_2) from the (a) and (b) scenario as "top pairs" because only there the relation between genes changes between classes. However, the pair from the scenario (b) should not be considered as a biological switch due to small change of the genes expression level between classes. Unfortunately, the undoubtedly relevant pair from the scenario (c) will not be considered by any currently available RXA-family algorithms despite significant variations in the expression values of genes in normal and cancer classes. It might choose them together with other genes, by making multiple top pairs, but besides potential interpretability problems, lower accuracy issues may also arise. Evo-REDT solution is capable not only of selecting relevant pairs (scenario (a) and (c)) but also ignoring the ones with small weight perturbations.

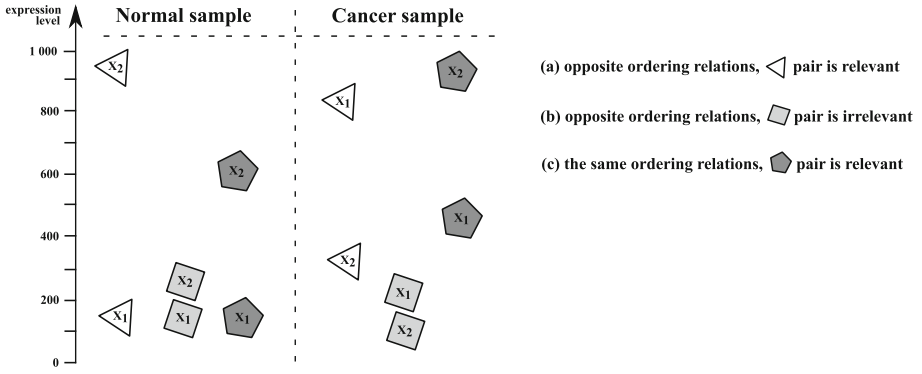

Fig. 1. Possible relations between two genes X1 and X2 in normal and cancer sample together with biological importance of the pair constituted from that genes

Additionally, RXA enormous computational complexity strongly limits the number of features and inter-relations that can be analyzed [13]. For regular RXA exhaustive search, it equals $O(T * M * N^2)$, where T is the number of splitting nodes of DT, M is the number of instances and N is the number of analyzed genes. Evo-REDT has much higher complexity due to additional search for the relations weight. For this newly arisen level of complexity, even a standard evolutionary approach might be not sufficient.

3 Evolutionary Relative Expression Decision Tree

The proposed solution has been integrated into a system called the Global Decision Tree (GDT). Its overall structure is based on a typical evolutionary

algorithm (EA) schema [17] with an unstructured population and generational selection. The GDT framework [15] can be used to induce various types of trees and its applications also cover biomedical data [6]. We have proposed several changes in the original GDT solutions, involving the node representation and overall evolutionary search. The general flowchart of the Evo-REDT solution is illustrated in Fig. 2.

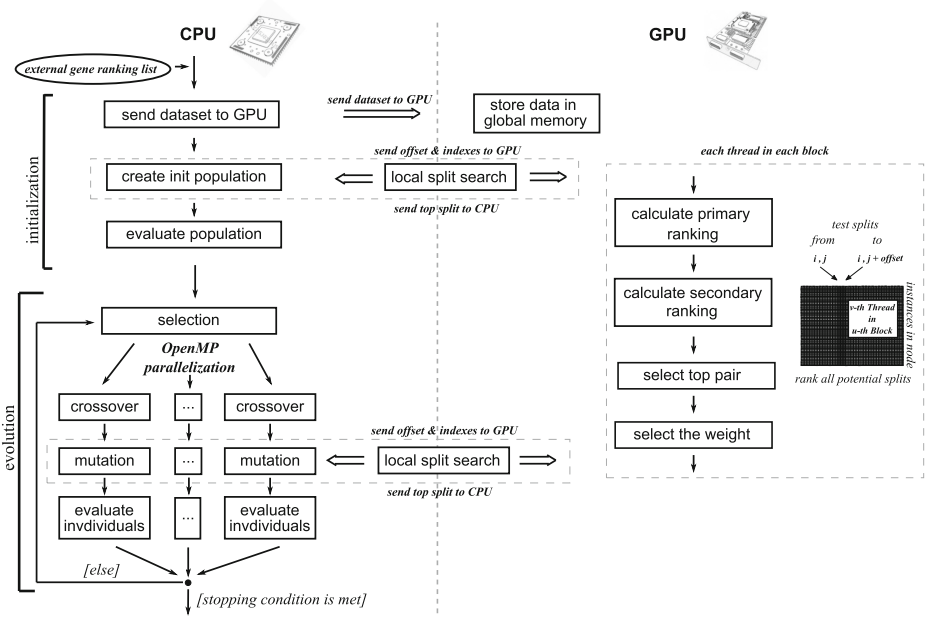

Fig. 2. General flowchart of the Evo-REDT solution

3.1 Representation, Initialization, Selection

Decision trees are quite complicated structures, in which a number of nodes, type of the tests and even number of test outcomes are not known in advance. The GDT system uses a tree-encoding schema in which individuals are represented in their actual form as potential tree-solutions. A new type of tests in the splitting nodes is applied. It is constituted from a single pair of genes together with the weight and has the form $(x_i > w * x_j)$. Additionally, each node stores information about training instances related to the node. This allows the algorithm to perform more effectively local modifications of the structure and tests during the application of genetic operators. Finally, we have embedded information about the discriminative power of genes calculated by the external tool (algorithm Relief-F was used [18]) in a form of ranked list. It is submitted as an

additional input to Evo-REDT and can be manually modified, for example, to focus on biomarker genes for a given disease.

In the GDT system, to maintain a balance between exploration and exploitation, initial individuals are created by using a simple top-down algorithm with randomly selected sub-samples of original training data. Before initialization, the dataset is first copied from the CPU main memory to the GPU device memory so each thread block can access it (see Fig. 2). It is performed only once before starting the tree induction as later only the indexes of the instances that are located in a calculated node are sent.

The selection mechanism is based on a ranking linear selection [17] with the elitist strategy, which copies the best individual founded so far to the next population. Evolution terminates when the fitness of the best individual in the population does not improve during a fixed number of generations (default: 100) or a maximum number of generations is reached (default: 1000).

3.2 Genetic Operators

To preserve genetic diversity, the GDT system applies two specialized genetic meta-operators corresponding to the classical mutation and crossover. Both operators may have a two-level influence on the individuals as either decision tree structure or a test in the splitting node can be modified. Depending on the position in the tree, different aspects are taken into account to determine the crossover or mutation point. If the change considers the overall structure, the level of the tree is taken into account. The modification of the top levels is performed less frequently than the bottom parts as the change would have a much bigger, global impact. The probability of selection is proportional to the rank in a linear manner. Examples of such variants are adding/deleting a node in the case of mutation and tree-branch crossover.

If the change considers the tests in the splitting nodes their quality is taken into account like the ones with the higher error, per instance, are more likely to be changed. In the case of mutation, it can be replacing a pair of genes with a new one or changing a single gene in a test. The first two variants require updating the weight between two genes that constitute a test. Additionally, in both variants, we use the gene ranking that determines which new genes will appear in the test. This way top genes from the dataset are considered more often in the population. Crossover variants allow whole tests to exchange as well as randomly selected genes from the pairs between the individuals.

3.3 Fitness Function

DTs are at some extent prone to overfitting [14]. In typical top-down induction, this problem is partially mitigated by performing a stop condition and applying post-pruning. In the case of evolutionary induced DT, this problem may be controlled by a multi-objective fitness function in order to maximize the accuracy and minimize the complexity of the output tree. In this work, we decided to use

a simple weight formula, but measure the tree complexity in a different way. The Evo-REDT system maximizes the following fitness function:

$$Fitness(T) = Q(T) - \alpha * Rank(T), \tag{4}$$

where: $Q(T)$ is the accuracy calculated on the training set, $Rank(T)$ is the sum of the ranks of attributes constituting tests and α is the relative importance of the complexity term (default value is 0.05) and a user supplied parameter. As we can see, instead of using the number of leaves or nodes, we measure the sum of the ranks of the attributes that constitute the tests in the internal nodes provided by the external Relief-F algorithm. This way the attributes with the higher rank are more likely to be used in the prediction model.

3.4 Parallelization

The GDT system supports various parallelization techniques [5,15]. However, in the context of biomedical data mining where the number of instances is low, using only the data-parallel decomposition strategy will not be effective [12]. We propose a hybrid approach with shared address space (OpenMP) paradigm and graphics processing units (GPU)-based parallelization. The individuals from the population are spread over the CPU cores using OpenMP threads. Each OpenMP thread is reponsible for subsequent algorithm blocks (genetic operator, evalution, etc.) for the assigned pool of individuals. This way, the individual are processed in parallel on the CPU.

The GPU parallelization is applied in a different way. When the mutation operator updates or calculates a new test in a splitting note, a local search for the top gene pair is performed. Each thread on the device is assigned an equal amount of relations (called offset) to compute so it 'knows' which relations of genes it should analyze and where it should store the result. However, finding a relation $x_i > w * x_j$ for a given set of instances that reached a particular node is still computationally demanding. That is why the first attribute is selected by the CPU which together with offset and indexes to the instances are sent to the GPU. Each thread in each block calculates the primary ranking which involves the number of times the relation holds in one of the classes and not in another one. The secondary ranking is a draw breaker, which is based on the differences in the weight relations in each class and object. The weight w of the top pair equals to x_i/x_j of the instance in which relation simultaneously distinguishes the instances from different classes and is the lowest among the instances from the same class. The weight can also be smoothed to e.g. a single precision value or even rounded to an integer in order to improve comprehensibility and at some extent the overall generalization (default: 0.5). After all block threads finished, the results are copied from the GPU device memory back to the CPU main memory and sorted according to the rank. Simplified ranking linear selection is used to select the pair of genes that will constitute the test in the splitting node.

4 Experimental Validation

Experimental analysis to evaluate the relative performance of the proposed approach is performed using several cancer-related gene expression datasets. We confront the Evo-REDT with popular RXA extensions as well as outline other algorithm characteristics.

4.1 Inducers, Datasets and Settings

To make a proper comparison with the RXA algorithms, we use the same 8 cancer-related benchmark datasets that were tested with the EvoTSP solution [4]. Datasets are deposited in NCBI's Gene Expression Omnibus and summarized in Table 1. A typical 10-fold cross-validation is applied and following RXA algorithms are confronted:

- TSP, TST, and k-TSP were calculated with the AUERA software [8];
- EvoTSP results were taken from the publication [4];
- original TSPDT and Evo-REDT implementations are used.

Table 1. Details of gene expression datasets: abbreviation with name, number of genes and number of instances.

Datasets	Genes	Instances	Datasets	Genes	Instances
(a) GDS2771	22215	192	(e) GSE10072	22284	107
(b) GSE17920	54676	130	(f) GSE19804	54613	120
(c) GSE25837	18631	93	(g) GSE27272	24526	183
(d) GSE3365	22284	127	(h) GSE6613	22284	105

In all experiments, a default set of parameters for all algorithms is used in all tested datasets and the presented results correspond to averages of several runs. Evo-REDT uses recommended GDT settings that were experimentally evaluated and given in details in GDT framework description [15], e.g.: population size: 50, mutation rate 80%, crossover rate 20%.

Due to the performance reasons concerning other approaches, the Relief-F feature selection was applied and the number of selected genes was arbitrarily limited to the top 1000. Experiments run on the workstation equipped with Intel Core i5-8400 CPU, 32 GB RAM, and NVIDIA GeForce GTX 1080 GPU card (8 GB memory, 2 560 CUDA cores). The sequential algorithm was implemented in C++ and the GPU-based parallelization part was implemented in CUDA-C (compiled by nvcc CUDA 10; single-precision arithmetic was applied).

Table 2. Inducers accuracy and size comparison, best for each dataset is bolded

Dataset	TSP	TST	k-TSP		EvoTSP		TSPDT		Evo-REDT	
	Acc.	Acc.	Acc.	Size	Acc.	Size	Acc.	Size	Acc.	Size
(a)	57.2	61.9	62.9	10	65.6	4.0	60.1	15.4	**72.9** ± 8.0	8.2 ± 1.1
(b)	88.7	89.4	90.1	6.0	96.5	2.1	**98.2**	1.0	**98.2** ± 5.7	2.2 ± 0.4
(c)	64.9	63.7	67.2	10	**78.1**	2.8	72.3	5.8	76.2 ± 9.9	7.3 ± 1.4
(d)	93.5	92.8	94.1	10	**96.2**	2.1	88.3	2.0	94.2 ± 8.8	2.8 ± 0.9
(e)	56.0	60.5	58.4	14	66.9	3.1	68.1	4.7	**73.0** ± 10.9	6.0 ± 0.8
(f)	47.3	50.1	56.2	18	66.2	2.7	67.2	10.9	**74.3** ± 6.2	7.9 ± 1.0
(g)	81.9	84.2	87.2	14	86.1	4.1	88.6	3.3	**91.5** ± 8.5	3.9 ± 0.7
(h)	49.5	51.7	55.8	10	53.6	6.1	59.6	7.0	**70.5** ± 16.9	8.4 ± 1.0
Average	67.4	69.3	71.5	11.5	76.2	2.7	75.3	6.2	**81.3** ± 9.4	5.8 ± 0.9

4.2 Accuracy Comparison of Evo-REDT to Popular RXA Counterparts

Table 2 summarizes classification performance for the proposed solution and its competitors. The model size of TSP and TST is not shown as it is fixed and equals correspondingly 2 and 3. Both, the evolutionary TSP approach called EvoTSP, as well as a top-down induced RXA decision tree TSPDT, are outperformed by the proposed Evo-REDT solution. The statistical analysis of the obtained results using the Friedman test and the corresponding Dunn's multiple comparison test (significance level/p-value equals 0.05), as recommended by Demsar [7] showed that the differences in accuracy are significant. We have also performed an additional comparison between the datasets with the corrected paired t-test with the significance level equals 0.05 and 9 degrees of freedom (n-1 degrees of freedom where n = 10 folds). It showed that Evo-REDT significantly outperforms all algorithms on more than half datasets. What is important, the trees induced by the Evo-REDT are not only accurate but also relatively small and simple. This indicates that the model managed to find more deep interaction and sub-interaction between the genes.

4.3 Evo-REDT Characteristics

To improve the overall generalization of Evo-REDT as well as the model comprehensibility, we have checked how rounding the weight relation between the genes impacts the results. Experimental results showed that there were no statistical differences between algorithms with 0.1, 0.5 respectively, and without rounding weights. Therefore, in Evo-REDT we used a default 0.5 rounding for the weight relation. An example of tree induced for the first dataset (GDS2771) is illustrated in Fig. 3. We can observe, that Evo-REDT found splitting pairs with various weights and the induced tree is small and easily interpretable.

In this section, we would also like to share some of the preliminary results to verify if the trees induced by the Evo-REDT are somehow useful. By using

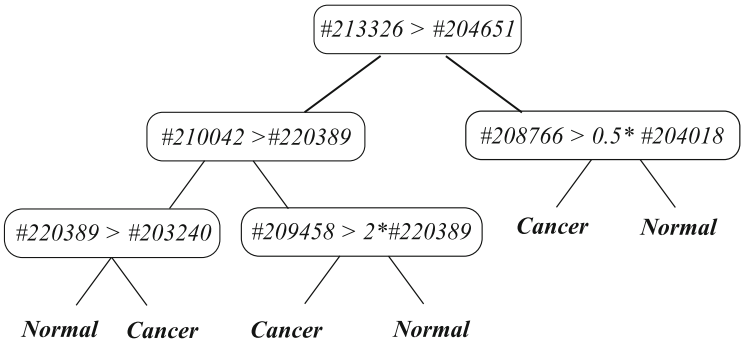

Fig. 3. An example decision tree induced by Evo-REDT with rounded to 0.5 weights for lung cancer data (GDS2771)

the GDS2771 dataset description available on GenBank NCBI [2] we performed a brief examination of our predictor (see Fig. 3). To check if genes found in the splitting nodes have some biological meaning we have decoded gene names from GDS2771 with GPL96 platform provided by NCBI (in the Figure genes are encoded as Affymetrix Probe Set ID). We found out that 2 out of 9 genes are directly related to lung cancer, another 2 were discussed in several papers while the remaining 5 were also visible in the medical literature. This is only an example of a fraction of knowledge discovered by Evo-REDT but even the presented model is at some point supported by biological evidence in the literature.

Much effort in this paper was put into improving the speed of the proposed solution. Table 3 shows the average calculation time for a single dataset without any parallel calculations and with OpenMP and/or GPU enabled. We also include the approximate induction time of other algorithms (if provided) for illustration purposes only. We cannot compare the results as the machines, software, etc. may be significantly different. However, with additional embedded feature ranking we managed to improve the EA convergence and reduce the number of required iterations which equals 1000 whereas for EvoTSP it is 10 times higher.

As expected, the sequential version of the algorithm is much slower than the rest of the Evo-REDT variants from Table 3. It should be noted that GPU-accelerated Evo-REDT may be applied to much larger gene expression datasets without any feature selection. The potential of the GPU parallelization was not fully utilized within performed experiments due to the limited number of features.

Table 3. Average time in seconds for the algorithm to train a model

Algorithm	Evo-REDT			TSP	TST	TSPDT	EvoTSP
	Seq.	OpenMP	OpenMP+GPU				
Time	637	171	110	2.1	712	152	2700

5 Conclusions

Finding simple decision rules with relatively high prediction power is still a major problem in biomedical data mining. Our new approach called Evo-REDT tackles this problem with a more generic approach of finding fractional relative relations between the genes. The proposed solution is composed of evolutionary DT inducer and extended concept of RXA. Our implementation covers multiple optimizations including OpenMP and GPU parallelizations as well as incorporates knowledge about the discriminative power of genes into the evolutionary search. Performed experiments show that the knowledge discovered by Evo-REDT is accurate, comprehensible and the model training time is relatively short.

We see many promising directions for future research. In particular, we are currently working with biologists and bioinformaticians to better understand the gene relations generated by Evo-REDT. Next, there is still a lot of ways to extend the tree representation e.g. by using more than one pair of genes in the splitting nodes. Optimization of the approach can also be improved e.g. load-balancing of tasks based on the number of instances in each node, simultaneous analysis of two branches, better GPU hierarchical memory exploitation. Finally, we want to validate our approach using proteomic and metabolomic data as well as integrated multi-omics datasets.

Acknowledgments. This project was funded by the Polish National Science Center and allocated on the basis of decision 2019/33/B/ST6/02386 (first author). The second and third author were supported by the grant WZ/WI-IIT/3/2020 from BUT founded by Polish Ministry of Science and Higher Education.

References

1. Bacardit, J., et al.: Hard data analytics problems make for better data analysis algorithms: bioinformatics as an example. Big Data **2**(3), 164–176 (2014)
2. Benson, D.A., et al.: GenBank. Nucleic Acids Res. **46**(D1), D41–D47 (2018)
3. Czajkowski, M., Kretowski, M.: Top scoring pair decision tree for gene expression data analysis. Adv. Exp. Med. Biol. **696**, 27–35 (2011)
4. Czajkowski, M., Kretowski, M.: Evolutionary approach for relative gene expression algorithms. Sci. World J. 593503 (2014). Hindawi
5. Czajkowski, M., Jurczuk, K., Kretowski, M.: A parallel approach for evolutionary induced decision trees. MPI+OpenMP implementation. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2015. LNCS (LNAI), vol. 9119, pp. 340–349. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19324-3_31
6. Czajkowski, M., Kretowski, M.: Decision tree underfitting in mining of gene expression data. An evolutionary multi-test tree approach. Expert Syst. Appl. **137**, 392–404 (2019)
7. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
8. Earls, J.C., et al.: AUREA: an open-source software system for accurate and user-friendly identification of relative expression molecular signatures. BMC Bioinform. **14**, 78 (2013)

9. Eddy, J.A., Sung, J., Geman, D., Price, N.D.: Relative expression analysis for molecular cancer diagnosis and prognosis. Technol. Cancer Res. Treat. **9**(2), 149–159 (2010)
10. Geman, D., et al.: Classifying gene expression profiles from pairwise mRNA comparisons. Stat. Appl. Genet. Mol. Biol. **3**(19) (2004)
11. Huang, X., et al.: Analyzing omics data by pair-wise feature evaluation with horizontal and vertical comparisons. J. Pharm. Biomed. Anal. **157**, 20–26 (2018)
12. Jurczuk, K., Czajkowski, M., Kretowski, M.: Evolutionary induction of a decision tree for large scale data. A GPU-based approach. Soft Comput. **21**, 7363–7379 (2017)
13. Kagaris, D., Khamesipour, A.: AUCTSP: an improved biomarker gene pair class predictor. BMC Bioinform. **19**(244) (2018)
14. Kotsiantis, S.B.: Decision trees: a recent overview. Artif. Intell. Rev. **39**(4), 261–283 (2013)
15. Kretowski, M.: Evolutionary Decision Trees in Large-Scale Data Mining. Studies in Big Data, vol. 59. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-21851-5
16. Magis, A.T., Price, N.D.: The top-scoring 'N' algorithm: a generalized relative expression classification method from small numbers of biomolecules. BMC Bioinform. **13**(1), 227 (2012)
17. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3rd edn. Springer, Heidelberg (1996). https://doi.org/10.1007/978-3-662-03315-9
18. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. Mach. Learn. **53**(1–2), 23–69 (2003)
19. Tan, A.C., Naiman, D.Q.: Simple decision rules for classifying human cancers from gene expression profiles. Bioinformatics **21**, 3896–3904 (2005)