TIGER: an evolutionary search for Top Inter-GEne Relations

Marcin Czajkowski*

Faculty of Computer Science,
Bialystok University of Technology,
Wiejska 45a, 15-351 Białystok, Poland
Email: m.czajkowski@pb.edu.pl
*Corresponding author

Anna Czajkowska

Department of Biotechnology,
Medical University of Bialystok,
Kilinskiego 1, 15-089 Białystok, Poland
Email: aczajkowska12@student.umb.edu.pl

Marek Kretowski

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

Abstract: Relative Expression Analysis (RXA) plays an important role in biomarker discovery and microarray data classification. It focuses on ordering relationships between the expression of small sets of genes rather than their raw values. Most of the RXA algorithms are preceded by feature selection as analysing all possible subsets of genes is computationally infeasible. In this paper, we propose an efficient solution that unifies major variants of RXA algorithms and is capable of searching top inter-gene relations even in large microarray datasets. A specialised evolutionary algorithm that incorporates and exploits knowledge about RXA into the evolutionary search allows exploring solution space with all available genes. By embedding information about the genes' discriminative power we managed to speed up the evolutionary process and to search for complex interactions between genes. Experimental validation shows that the proposed solution outperforms popular RXA algorithms and has considerable potential for discovering new relationships between the genes.

Keywords: relative expression analysis; top-scoring pair; microarray classification; evolutionary algorithm; embedded feature selection.

Reference to this paper should be made as follows: Czajkowski, M., Czajkowska, A. and Kretowski, M. (2016) 'TIGER: an evolutionary search for Top Inter-GEne Relations', *Int. J. Data Mining and Bioinformatics*, Vol. 16, No. 2, pp.170–182.

Biographical notes: Marcin Czajkowski received his Masters degree (2007) and his PhD with honours (2015) in Computer Science from the Bialystok

University of Technology, Poland. His research activity mainly concerns machine learning and data mining, in particular, classification and regression trees and evolutionary algorithms

Anna Czajkowska received her Master in Pharmacy degree from Medical University of Bialystok in 2010; and she is a PhD student in Pharmacy at Medical University of Bialystok. Her current research interests are genomics, anticancer drugs and computer-aided drug design.

Marek Kretowski received his Masters degree in Computer Science from the Bialystok University of Technology, Poland in 1996. His PhD thesis defended in 2002 was prepared in a framework of collaboration between Laboratory of Signal and Image Processing, University of Rennes 1, France and Faculty of Computer Science, Bialystok University of Technology. In 2009, he received his DSc (Habilitation) in Computer Science from Institute of Computer Science, Polish Academy of Science. Currently, he works as an Associate Professor in Faculty of Computer Science, Bialystok University of Technology. His current research focuses on data mining methods and biomedical applications of computer science.

1 Introduction

With the rapid growth and the popularity of microarray technology a large amount of gene expression datasets became publicly accessible (Taminau et al., 2012). Availability of this information opens new challenges for existing algorithms that search for the relations between the genes. Finding accurate and simple rules or biomarker genes in whole gene expression dataset is still a real challenge and requires new efficient and robust classification algorithms. In the literature, we may find a good number of supervised machine learning algorithms. Among the most popular ones, we could mention the support vector machines (SVMs), neural networks, K-nearest neighbours or decision trees. Most of methods provide 'black box' decision rules that usually involve many genes combined in a highly complex fashion and achieve high predictive performance. However, it can be observed that there is a strong need for 'white box', comprehensive classification models which may actually help in understanding and identifying casual relationships between specific genes (Barros et al., 2014; Czajkowski et al., 2014).

A Relative Expression Analysis (RXA) focuses on finding interactions among small group of genes and studies the relative ordering of their expression values. In the pioneer research (Geman et al., 2004), authors used ranks of genes instead of their raw expression values and introduced the Top Scoring Pair (TSP) concept. The classification algorithms based on that idea appeared robust to small perturbations of gene expression values and insensitive to data normalisation and standardisation procedures. They managed to identify many interesting gene-gene interaction and played important role in a biomarker discovery (Lin et al., 2009). The influence of RXA solutions could be even greater, however, the computational complexity of the algorithms strongly limits the number of genes that can be analysed.

To face this problem, we propose a new algorithm called TIGER, which stands for Top Inter-GEne Relations. In contrast to other RXA algorithms, TIGER can efficiently

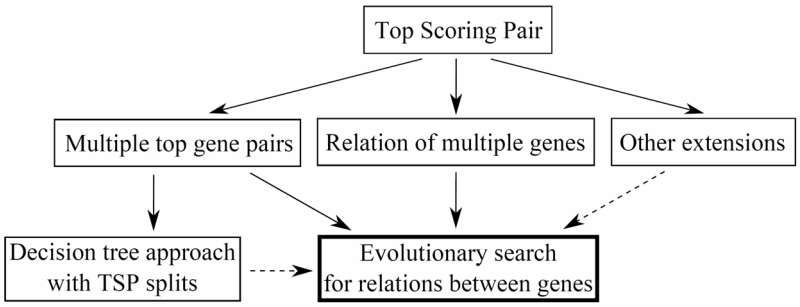
search for gene-gene interactions in full microarray datasets. Our specialised evolutionary algorithm (EA) not only combines and unifies major variants of TSP extensions but is capable of finding even larger and more complex inter-gene relations. The TIGER solution reviews and significantly extends other works that searched for top scoring gene pairs, especially the EvoTSP (Czajkowski and Kretowski, 2014) algorithm. We propose specialised variants of the genetic operators and modify the fitness function to improve the evolutionary process. To boost the TIGER's speed, we have designed a novel ranking of genes that is based on their discriminative power. The ranking is used to calculate the probability of selecting genes into the classifiers model. This solution allows to consider the relations based on top genes more often during the evolutionary process. Experimental validation illustrates the advantage of the proposed approach in comparison to predecessors.

The rest of the paper is organised as follows. The next section provides a brief background on the RXA algorithms. Section 3 describes our approach and Section 4 presents the experimental validation of TIGER and competitive algorithms on 8 real microarray datasets. In the last section, the paper is concluded and possible future works are sketched.

2 Algorithms for relative expression analysis

Gene expression data is very challenging for computational tools and mathematical modelling. Traditional solutions often fail due to the high ratio of features/observations as well as enormous genes redundancy. Therefore, the new computational tools are proposed to extract significant and meaningful rules from microarray data, and among them RXA algorithms are gaining popularity. Figure 1 illustrates the relative expression algorithms taxonomy that includes main development paths that will be now briefly described. The bolded frame indicates the location of our approach.

Figure 1 The general taxonomy of the family of top scoring pair algorithms with a bolded location of the TIGER algorithm

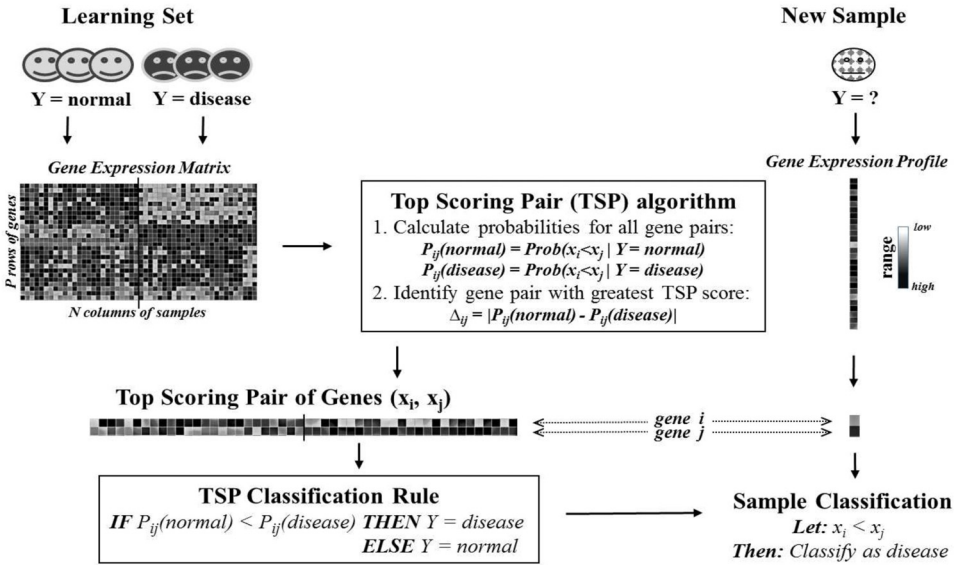


2.1 Algorithms

The first and the most popular RXA solution is the Top-Scoring Pair (TSP) proposed by Donald Geman (Geman et al., 2004). The algorithm bases on pairwise comparison of gene expression values and searches for a single pair of genes with the highest rank. The

general schema of the algorithm is illustrated in Figure 2. Let us assume that x_i and x_j are the expression values of two different genes from available set of genes and there are only two classes: *normal* and *disease*. At first, we calculate the probability of the relation $x_i < x_j$ between those two genes in the objects from the same class: $P_{ij}(normal) = Prob(x_i < x_j \mid Y = normal)$ and $P_{ij}(disease) = Prob(x_i < x_j \mid Y = disease)$, where Y denotes the class of the objects. Next, the score for this pair of genes (x_i, x_j) is calculated: $\Delta_{ij} = \mid P_{ij}(normal) - P_{ij}(disease) \mid$. This procedure is repeated for all distinct pairs of genes and the gene with the highest score becomes titled top scoring pair. In case of a draw, a secondary ranking that bases on genes expression differences in each class and object is used (Tan and Naiman, 2005). Finally, for the 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 the higher probability P_{ij} in the training set, otherwise it votes for the class with smaller probability.

Figure 2 The general schema of the top scoring pair algorithm



One of the first extensions of the TSP solution focused on increasing the number of pairs in the prediction model. The k-TSP algorithm (Tan and Naiman, 2005) applies no more than k top scoring disjoint gene pairs with the highest score, where the parameter k is determined by the internal cross-validation. This method was later combined with a top-down induced decision tree in an algorithm called TSPDT (Czajkowski and Kretowski, 2011). In this hybrid solution each non-terminal node of the tree divides instances according to a splitting rule that is based on TSP or k-TSP algorithm.

Alternative approach for the TSP extension searches for relationships between more than two genes. Top Scoring Triplet (TST) (Lin et al., 2009) and Top Scoring N (TSN) (Magis and Price, 2012) algorithms analyse various ordering relationships between the genes, however, the general concept of TSP is retained. There exist other extensions of

the TSP solution that focus on protein expression (Kau et al., 2012), work as a feature selection for more complex classifiers (Shi et al., 2011; Zhang et al., 2012) or integrate various volumes of microarray data (Lin et al., 2009).

The latest TSP extensions perform evolutionary search for different relations between the genes. In the GTSPDT algorithm (Czajkowski and Kretowski, 2013) authors propose hierarchical evolutionary method that extends TSPDT by performing a global induction of decision tree. Preliminary results showed that this evolutionary search may be a good alternative to the traditional RXA algorithms. The idea of applying EA for the search of TSP was later continued in an algorithm called EvoTSP (Czajkowski and Kretowski, 2014). The authors proposed specialised EA that combined different variants of the TSP solutions and allowed exploring larger solution space.

2.2 RXA limitations

One of the main drawback of RXA algorithms is high computational complexity that equals $O(k*Z^N)$, where k is the number of top-scoring groups, Z is the number of analysed genes and N is the size of group of genes which ordering relationships is searched. This slow performance is caused by the consideration of all possible gene pairs or gene groups and thus it limits the size of the group of genes that can be analysed. The largest ordering relationship was tested on a group of 4 genes (N=4) but only when the total number of analysed genes was heavily reduced by the feature selection to a few hundreds (Magis and Price, 2012). Although, the parallelisation of the algorithm managed to speed up calculation time by two orders of magnitude (Magis et al., 2011), it is still computationally infeasible to calculate on a full microarray dataset.

Second limitation of the RXA algorithms is the need of presenting the parameters k and N for the algorithms. It is almost impossible to define, for a particular problem in advance, what is the type of relationships in a dataset and how many genes or gene-pairs should be involved. For the k-TSP algorithm the parameter k is determined by the internal cross-validation which increases the calculation time and decreases the size of already small training set. It is also not clear which of the TSP solution should be applied: TSP, k-TSP, TSN or TSPDT and due to the computational complexity the potentially hybrid solutions like k-TSN or decision tree with TSN were never published.

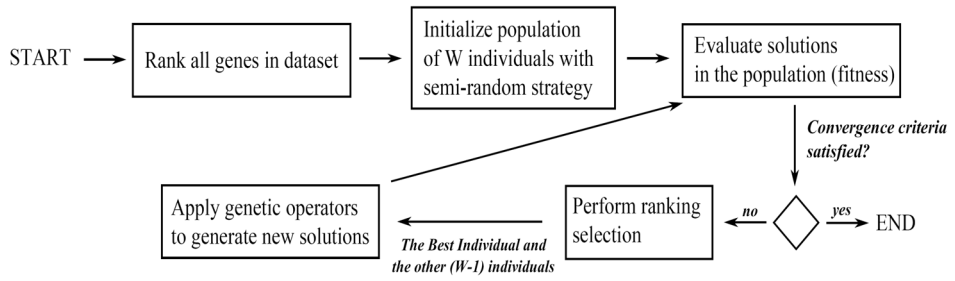
One of the ways to extend the search for more complex relations between the genes is application of some heuristic methods. The EvoTSP algorithm managed to limit aforementioned drawbacks of RXA algorithms through the evolutionary approach. Proposed specialised EA searches for the weight top scoring pairs and allows exploring larger solution space. Although, EvoTSP does not calculate all combinations of genes, it still requires high computation time especially when full microarray datasets are analysed. The EvoTSP algorithm treats all the genes with the same importance and does not consider their discriminative power. It causes slower convergence of the algorithm which, for most of the time, focuses on meaningless genes. Therefore, in our approach we propose to treat low and high rank genes differently and consider the latter ones more often in the evolutionary search. In addition, the EvoTSP model focuses on gene pairs whereas the TIGER algorithm searches for gene relations which are not limited to 2 genes.

3 Evolutionary search for Top Inter-GEne Relations

The evolutionary algorithms (EA) (Michalewicz, 1996) belong to a family of meta-heuristic methods that are inspired by biological mechanisms of evolution and represent techniques for solving a wide variety of difficult optimisation problems. The typical EA operates on a population of individuals that represents possible solutions to the target problem. In each evolutionary iteration, individuals are modified with genetic operators such as mutation and crossover, and evaluated according to the fitness function. Next, individuals are reproduced to a new population of offspring whereas individuals with higher fitness are reproduced more often. The evolutionary loop is stopped when the convergence criteria are satisfied.

In this section, we would like to propose the solution called Top Inter-GEne Relations (TIGER), which is an evolutionary approach for the RXA algorithms. The TIGER algorithm follows a framework for EA with an unstructured population and a generational selection (Michalewicz, 1996). The main steps of the solution are illustrated in Figure 3.

Figure 3 The TIGER process diagram



3.1 Ranking the genes

The TIGER algorithm requires at its input the rank of all genes in the analysed dataset. The knowledge of discrimination power of each gene is latter used in initialisation of the population as well as in different variants of genetic operators. Ranking can be performed with any algorithm that assigns ranks to each gene by some importance criterion. In our research we applied the Relief-F (Robnik-Siikonja and Kononenko, 2003) algorithm which is commonly used to do variable selection of microarray data. The list of ranked genes that is passed to the TIGER solution can also be modified manually if there is a need for some gene or group of genes to be rejected.

It should be noted that at this step no genes should be excluded from the dataset to let the TIGER solution work on all available genes. This way the algorithm is capable of finding interesting relations in low ranked genes which may constitute model with high discriminative power. This would be not possible, if the feature selection was applied as it has place in most of the studies.

3.2 Representation and initialisation

Associations between the genes may be represented by complicated structures in which the number of relations and the number of genes in each relation is not known in

advance. Therefore, in the TIGER system, relations between gens are not encoded in individuals and are represented in their actual form as a group of relations between the genes. Let us focus on an example that is illustrated in Figure 4. One can observe that the individual is composed of 3 different relations, where each one could appear in previous TSP extensions. However, the TIGER algorithm not only combines and unifies main TSP extensions, but also mitigates their limitations, like a restriction in k-TSP to use only disjoint gene pairs. With the evolutionary search of multiple relations between the genes each individual may have any type of relation from main RXA algorithms. The additional weight parameter denoted as r_i reflects the importance of the *i*-th relation. This way, when there are many relations in the model, the final decision is made by using the weight voting where each relation's vote equals to its r_i parameter.

Figure 4 An example representation of TIGER individual with 3 relations

TIGER: an evolutionary search for Top Inter-GEne Relations

r_i *- weight of the relation* $\rightarrow$

$\{\ r_1 \cdot (x_i > x_j),$ ← *corresponds to TSP/k-TSP*

$r_2 \cdot (x_i > x_k > x_t),$ ← *corresponds to TST/TSN*

$r_3 \cdot (x_m \leq x_n)\ \}$ ← *TSP inverted relation*

Traditionally, the initial population should be generated randomly to cover the entire range of possible solutions. In addition, the direct application of one of the TSP algorithms can trap the EA in a local optima. Therefore, while creating the initial population, we search for a good trade-off between a high degree of heterogeneity and a relatively low computation time. Each initial individual contains no more than 5 relations – each 2 genes long. Due to the large set of possible genes, we used exponential ranking selection (Blickle and Thiele, 1995) to the list of ranked genes that is passed to the TIGER algorithm. With this strategy, it is more likely to have relations between the genes that have high discriminative power.

3.3 Genetic operators

Two specialised genetic meta-operators corresponding to classical mutation and crossover have been proposed to maintain the genetic diversity. We have applied and extend basic variants from the EvoTSP solution and propose new ways to diversify relations in the individuals. The crossover variants include:

- a random chosen relation is exchanged between two affected individuals;
- genes within the relations are exchanged between the individuals;
- random relations from the best individual founded so far are added to the affected individual;
- random relations from the best individual founded replaces random relation in affected individual.

Set of mutation variants in the proposed solution covers:

- add or remove a new relation in the affected individual;
- change relation in the individual by replacing the gene or switching the relation sign between genes;
- increase or decrease the weight of the relation;

One of the TIGER's innovative solutions is the use of gene rankings in different variants of mutation operator. Alike in the initial population procedure, exponential ranking selection determines which new genes (or relations) will appear in the model. This way top genes from the dataset are considered more often in the population, but the low-ranked genes can still appear in the model.

3.4 Fitness function, selection and terminal condition

The fitness function is one of the most important and sensitive elements in the design of the EA. It measures how good a single individual is in terms of meeting the problem objective. As direct minimalisation of classification error usually leads to the over-fitting problem, the multi-objective optimisation may present more acceptable overall results (Cawley and Talbot, 2010). The TIGER solution adapts the idea proposed in the CART system (Breiman et al., 1984) and also used in the EvoTSP algorithm. However, in contrast to EvoTSP the complexity term focuses only on the number of unique genes that appear in classification model rather than the total number of top gene pairs. The fitness function has the following form:

$$Fitness_{individual} = Accuracy - \alpha * Complexity,$$

where $Accuracy$ is the reclassification quality calculated on the training set and $Complexity$ equals to the total number of unique genes that compose relations in the individual's model. The $alpha$ parameter can be viewed as the relative importance of the complexity term. It can be specified by the user to steer the output model complexity and to tune the classifier to the currently analysed dataset.

Ranking linear selection is applied as a selection mechanism. In Figure 3 we can see that in each iteration, the single individual with the highest value of fitness function in the 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: 1000). However, in case of a slow convergence, the maximum number of generations (default: 10000) is also specified to limit the computation time.

4 Results and discussion

We have performed experiments on several publicly available microarray datasets to verify the TIGER algorithm prediction power. In the experiments, the performance of the proposed solution with respect to the classification accuracy and size of the model is confronted with popular RXA algorithms.

4.1 Datasets and setup

In order to make a proper comparison with the RXA algorithms, we have selected benchmark datasets that were used in testing the EvoTSP solution (Czajkowski and Kretowski, 2014). Eight publicly available microarray datasets related to human problems deposited in NCBI's Gene Expression Omnibus (Edgar et al., 2002) and summarised in Table 1 were used. We have precisely followed the preprocessing and the experiments procedure to make the comparison to the results enclosed in the paper as accurate as possible. A typical tenfold cross-validation was used and the testing of different RXA algorithms was performed with the AUERA software (Earls et al., 2013), which is an open-source system for identification of relative expression molecular signatures. 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 20 runs.

Table 1 Details of tested gene expression datasets – the dataset names with abbreviation, number of genes and number of instances

Datasets (DT)	*Abbreviation*	*Genes*	*Instances*	*Description*
GDS2771	A	22215	192	Lung cancer
GSE17920	B	54676	130	Hodgkin lymphoma
GSE25837	C	18631	93	Chronic loneliness
GSE3365	D	22284	127	Inflammatory Bowel disease
GSE10072	E	22284	107	Lung adenocarcinoma
GSE19804	F	54613	120	Lung cancer
GSE27272	G	24526	183	Impact of tobacco smoke
GSE6613	H	22284	105	Parkinson's disease

4.2 Comparison of TIGER to other RXA algorithms

We have selected main RXA algorithms to compare with proposed TIGER solution: TSP, k-TSP, TSN and EvoTSP. The AUREA software sets the maximum number of top-scoring pairs (parameter k) for k-TSP to 10 and N in TSN algorithm is set by default to 3. No feature selection was performed for EvoTSP and TIGER algorithms, however, for all other solutions the AUERA software needed a feature selection step, because of the computational complexity of the algorithms.

4.3 Comparison of top-scoring family algorithms methods

Table 2 summarises classification performance for the proposed solution EvoTSP and it's competitors: TSP, TSN, k-TSP. The model size of TSP and TSN is not shown as it is fixed and equals correspondingly 2 and 3.

The results enclosed in Table 2 show that the TIGER solution can successfully compete with popular RXA algorithms. The statistical analysis of the obtained results using the Friedman test and the corresponding Dunn's multiple comparison test (significance level equals 0.05), as recommended by Demsar (2006) showed that the TIGER solution significantly outperforms all tested RXA algorithms. We have also performed additional comparison between the datasets with corrected paired t-test

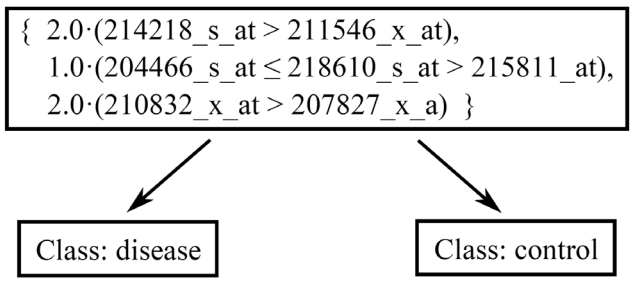
(Nadeau and Bengio, 2003) with the significance level equals 0.05 and 9 degrees of freedom ($n-1$ degrees of freedom where $n=10$ folds). It showed that TIGER significantly outperforms TST, TSN and k-TSP algorithms on all datasets except the dataset D where there are no statistical differences between the classifiers accuracy. Number of genes that constitute the TIGER model is similar to the EvoTSP solutions and significantly smaller than in k-TSP.

Table 2 Comparison of top-scoring algorithms, including accuracy with its standard deviation and the number of unique genes that build classifier's model. The highest classifiers accuracy for each dataset was bolded

DT	*TSP*	*TSN*	*k-TSP*		*EvoTSP*		*TIGER*	
	Accuracy	*Accuracy*	*Accuracy*	*Size*	*Accuracy*	*Size*	*Accuracy*	*Size*
A	57.2 ± 2.4	61.9 ± 2.8	62.9 ± 3.3	10	65.6 ± 2.0	4.0	**72.7** ± 3.6	2.8
B	88.7 ± 2.6	89.4 ± 2.1	90.1 ± 2.5	6	96.5 ± 1.3	2.1	**97.4** ± 0.6	2.0
C	64.9 ± 3.5	63.7 ± 4.7	67.2 ± 3.2	10	**78.1** ± 2.6	2.8	78.0 ± 3.5	3.1
D	93.5 ± 1.7	92.8 ± 1.5	94.1 ± 1.6	10	**96.2** ± 1.1	2.1	93.0 ± 1.6	2.0
E	56.0 ± 4.0	60.5 ± 5.1	58.4 ± 4.0	14	66.9 ± 5.6	3.1	**75.9** ± 2.8	4.0
F	47.3 ± 4.8	50.1 ± 3.8	56.2 ± 2.2	18	**66.2** ± 1.1	2.7	65.4 ± 1.9	3.0
G	81.9 ± 2.6	84.2 ± 2.7	87.2 ± 2.1	14	86.1 ± 2.8	4.1	**89.5** ± 2.1	3.0
H	49.5 ± 3.5	51.7 ± 2.8	55.8 ± 5.3	10	53.6 ± 5.4	6.1	**64.4** ± 4.0	5.3
Avg.	**67.4 ± 3.3**	**69.3 ± 3.2**	**71.5 ± 3.0**	**11.5**	**76.2 ± 2.7**	**3.4**	**79.5 ± 2.5**	**3.1**

In Figure 5 we present one of the classification models with real gene names generated for one of the cross-validation folds of Parkinson's disease dataset (GSE6613) by the TIGER solution. In the example, the TIGER's classification model is constituted by 3 relations with total number of 7 unique genes. We can observe that the first and third relation are the typical TSP models or a single k-TSP model with $k=2$. The second relation is similar to one of the variants of the TST model. For this particular example, few genes from Figure 5 are actually related to the SNCA gene and are connected to the Parkinson's disease brain (Lewis and Cookson, 2012).

Figure 5 Output for Parkinson's disease dataset (GSE6613) for the TIGER algorithm



As for the direct confrontation between TIGER and EvoTSP, the proposed solution significantly outperforms EvoTSP on 4 datasets (A, E, G, H) and is significantly worse only on a single dataset (D). On the rest of the datasets there were no statistically significant differences. The sizes of the models are similar which suggests that the

TIGER algorithm is able to find better relations using the same number of genes than EvoTSP. Another important difference is the training time of the classifiers. Due to the use of gene ranking in TIGER the number of necessary evolutionary iterations decreased on average 5 times in comparison to EvoTSP. Performed experiments showed that, depending on the dataset, the TIGER solution needs from several seconds to a few minutes on a typical PC (Intel Core I5, 4 GB RAM) to build a classification model. It is longer than for RXA algorithms tested with AUREA software which usually needed up to a minute to build a model. However, it should be noted that TIGER performs the search on all available genes where AUERA software requires a feature selection. Without it, the algorithms would have to check all combinations of pairs or triples which would take several orders of magnitude more time than for proposed solution. In addition, TIGER solution is 2-times faster (in average on all datasets) than its ancestor EvoTSP due to the faster convergence of the evolutionary algorithm.

5 Conclusion

Existing RXA algorithms are simple, white box solutions that have relatively high prediction power on gene expression data. One of their main drawbacks is the calculation time which imposes on the algorithms many restrictions and a need of feature pre-selection. In this paper, we introduce effective classification tool that combines the power of EA and relative expression algorithms. Thanks to the specialised genetic operators and a multi-objective fitness function the TIGER algorithm is capable of finding complex relations between the genes. The efficiency of the solution is achieved by embedding additional information about the discriminative power of the genes in the evolutionary process. This way the TIGER algorithm does not need any feature selection and allows exploring much larger solution space. Experimental validation showed that the TIGER solution outperform all other RXA algorithms in context of the prediction power and speed.

We see many promising directions for future research. In particular, we could study more deeply the biological aspect of the rules generated by proposed system. We also plan to extend the possibility of the TIGER solution to work with unbalanced and multiclass microarray datasets. Finally, we consider adapting the proposed classification algorithm to work with protein expression databases.

References

Barros, R.C., Basgalupp, M.P., Freitas, A.A. and Carvalho, A.F. (2014) *Evolutionary Design of Decision-Tree Algorithms Tailored to Microarray Gene Expression Data Sets. IEEE Transactions on Evolutionary Computation*, Vol. 18, No. 6, pp.873–892.

Blickle, T. and Thiele, L. (1995) 'A comparison of selection schemes used in genetic algorithms', *TIK-Report*, Vol. 11, No. 2.

Breiman, L., Friedman, J., Olshen, R. and Stone C. (1984) *Classification and Regression Trees*, Wadsworth Int. Group.

Cawley, G.C. and Talbot, N.L.C. (2010) 'On over-fitting in model selection and subsequent selection bias in performance evaluation', *Journal of Machine Learning Research*, Vol. 11, pp.2079–2107.

Czajkowski, M. and Kretowski, M. (2011) '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.

Czajkowski, M. and Kretowski, M. (2013) 'Global top-scoring pair decision tree for gene expression data analysis', *Proceedings of EuroGP'12, Lecture Notes in Computer Science*, Vol. 7831, pp.229–240.

Czajkowski, M. and Kretowski, M. (2014) 'Evolutionary approach for relative gene expression algorithms', *The Scientific World Journal*, Hindawi 593503.

Czajkowski, M., Grzes, M. and Kretowski, M. (2014) 'Multi-test decision tree and its application to microarray data classification', *Artificial Intelligence in Medicine*, Vol. 61, No. 1, pp.35–44.

Demsar, J. (2006) 'Statistical comparisons of classifiers over multiple data sets', *Journal of Machine Learning Research*, Vol. 7, pp.1–30.

Earls, J.C., Eddy, J.A., Funk, C.C., Ko, Y., Magis, A.T. and Price, N.D. (2013) 'AUREA: an open-source software system for accurate and user-friendly identification of relative expression molecular signatures', *BMC Bioinformatics*, Vol. 14, No. 78.

Edgar, R., Domrachev, M. and Lash, M.E. (2002) 'Gene expression omnibus: NCBI gene expression and hybridization array data repository', *Nucleic Acids Research*, Vol. 30, No. 1, pp.207–210.

Geman, D., d'Avignon, C., Naiman, D.Q. and Winslow, R.L. (2004) 'Classifying gene expression profiles from pairwise mRNA comparisons', *Statistical Applications in Genetics and Molecular Biology*, Vol. 3, No. 19.

Kau, P., Schlatzer, D., Cooke, K. and Chance, M.R. (2012) 'Pairwise protein expression classifier for candidate biomarker discovery for early detection of human disease prognosis', *BMC Bioinformatics*, Vol. 13, No. 191.

Kretowski, M. and Grzes, M. (2007) 'Evolutionary induction of mixed decision trees', *International Journal of Data Warehousing and Mining*, Vol. 3, No. 4, pp.68–82.

Lewis, P.A. and Cookson, M.R. (2012) 'Gene expression in the Parkinson's disease brain', *Brain Research Bulletin*, Vol. 88, No. 4, pp.302–312.

Lin, K.M., Kang, J., Shin, H. and Lee, J. (2009) 'A cube framework for incorporating inter-gene information into biological data mining', *International Journal of Data Mining and Bioinformatics*, Vol. 3, No. 1, pp.3–22.

Lin, X., Afsari, B., Marchionni, L., Cope, L., Parmigiani, G., Naiman, D. and Geman, D. (2009) 'The ordering of expression among a few genes can provide simple cancer biomarkers and signal BRCA1 mutations', *BMC Bioinformatics*, Vol. 10, No. 256.

Magis, A.T. and Price, N.D. (2012) 'The top-scoring N algorithm: a generalized relative expression classification method from small numbers of biomolecules', *BMC Bioinformatics*, Vol. 13, No. 227.

Magis, A.T., Earls, J.C., Ko, Y., Eddy, J.A. and Price, N.D. (2011) 'Graphics processing unit implementations of relative expression analysis algorithms enable dramatic computational speedup', *Bioinformatics*, Vol. 27, No. 6, pp.872–873.

Michalewicz, Z (1996). 'Genetic algorithms + data structures = evolution programs', 3rd ed., Springer.

Nadeau, C. and Bengio, Y. (2003) 'Inference for the generalization error', *Machine Learning*, Vol. 52, pp.239–281.

Robnik-Siikonja, M. and Kononenko, I. (2003) 'Theoretical and empirical analysis of ReliefF and RReliefF', *Machine Learning*, Vol. 53, pp.23–69.

Shi, P., Ray, S., Zhu, Q. and Kon, M.A. (2011) 'Top scoring pairs for feature selection in machine learning and applications to cancer outcome prediction', *BMC Bioinformatics*, Vol. 12, No. 375.

Taminau, J., Meganck, S. et al. (2012) 'Unlocking the potential of publicly available microarray data using inSilicoDb and inSilicoMerging R/Bioconductor packages', *BMC Bioinformatics*, Vol. 13, No. 335.

Tan, A.C. and Naiman, D.Q. (2005) 'Simple decision rules for classifying human cancers from gene expression profiles', *Bioinformatics*, Vol. 21, pp.3896–3904.

Zhang, H., Wang, H, Dai, Z., Chen, M. and Yuan, Z. (2012) 'Improving accuracy for cancer classification with a new algorithm for genes selection', *BMC Bioinformatics*, Vol. 12, No. 298.