A Combined Genetic Programming for Microarray Data Analysis

K.Umamaheswari, Ph.D
Professor, Department of Information Technology
PSG College of Technology
Coimbatore, India

Dhivya.M
Assistant Professor, Department of Information Technology
SRM University
Chennai, India

Chithra.S
Student, Department of Information Technology
PSG College of Technology
Coimbatore, India

ABSTRACT
Microarray technology is a powerful tool to monitor gene expression or gene expression changes of hundreds or thousands of genes in a single experiment. Meta-Genetic Programming is the meta learning technique of evolving a genetic programming system to predict cancer classes for better understanding of different types of cancers and to find the possible biomarkers for diseases. A new technique which is known as Majority Voting Genetic Programming Classifier (MVGPC) combined with meta-genetic programming (MGP) is proposed which combines meta-genetic programming and majority voting technique to predict the cancer class for a given patient sample with higher accuracy and minimum computational time. This paper also aims to provide a means to identify cancer at an early stage and hence increase the chances of survival for the patients.

General Terms
Data Mining.

Keywords
Microarray, Meta-genetic programming, Majority voting, Feature ranking

1. INTRODUCTION
Due to the advanced developments in DNA microarray technology, scientists can now easily measure the expression levels of thousands of genes simultaneously in a biological organism. This has also made it possible to create databases of cancerous and normal tissues. Researchers have found evidence that there exist systematic differences in the gene expression levels of different tumor and normal tissues, and they are trying to correlate the clinical behavior of cancers with these differential gene expression patterns using various machine learning techniques.

2. META GENETIC PROGRAMMING
Machine learning techniques suffers from a problem known as over-fitting, where a large number of test samples needs to be classified when compared to the small number of training samples available to train the machine learning algorithm. Due to these problems, Genetic Programming (GP)[1], which is a part of evolutionary computing, was proposed for this correlation. Genetic programming is based on natural selection and evolution. The main advantage of GP over rank based methods like SNR[2] and other classifiers like support vector machine(SVM)[3] or k-nearest neighbor (kNN)[4] is that it can act as a classifier, as well as a gene selection algorithm. GP suffers from the drawback that it has to perform both classification and gene selection over a large set of data simultaneously leading to poor accuracy. Meta-genetic programming (MGP) [5] encodes the operators that act on the selected genes as trees. This variable length representation allows the simultaneous evolution of the operators along with the population of solutions.

3. MAJORITY VOTING GENETIC PROGRAMMING CLASSIFIER
This technique [6] works by combining several rules rather than just relying on a single rule. A group of rules can be more accurate[7] than the best member of the group and, multiple rules are evolved in different GP runs. Each rule is applied to a test sample, and count their votes with respect to each class. Then, the sample with highest number of votes is assigned to the class. However, the majority voting technique relies on the number of rules per voting group and on the rate of false prediction (on test samples) by single rules.

4. PROPOSED ALGORITHM(MVGPC+MGP)
Meta-genetic programming is used to create rules and then applied to training samples to evaluate their fitness. Based on fitness, best rules are selected and those rules are applied to the test samples to find the class of the test sample. The votes obtained in favor of each class are counted and the class with the maximum number of votes is considered as the winner class. This technique provides better results because it relies on several rules to make the decision rather than relying on a single rule. The algorithm for combining MVGPC and MGP is described in Fig.1.

```
1.  Collect the cancer data necessary for performing
    classification
2.  Divide the data into training and test subsets
3.  Generate a population of individuals by random
    composition of functions and gene expressions
4.  Evaluate the fitness of each of the individuals by
    applying it to the training sample
5.  Generate an operator by random composition of
    functions
6.  Select two individuals based on fitness and apply
    operator to the selected individuals
7.  Evaluate the fitness of the resultant offspring and
    select the best rule

8.  If the maximum number of generations is not over,
    then repeat steps 6 to 8. Else proceed to step 10
9.  If the maximum number of trial runs is not over, then
    repeat steps 5 to 9. Else proceed to step 11
10. Apply the best rules selected to the test samples
11. Count the number of votes in favor for each class
```

12. Set the class of the test sample as the class with the highest number of votes and Calculate accuracy and feature ranking

Fig 1: Algorithm for MVGPC+MGP

In gene-expressions based classification, the individuals in a GP population are S-expressions of classification rules consisting of functions and terminals corresponding to the genes of a microarray data set.

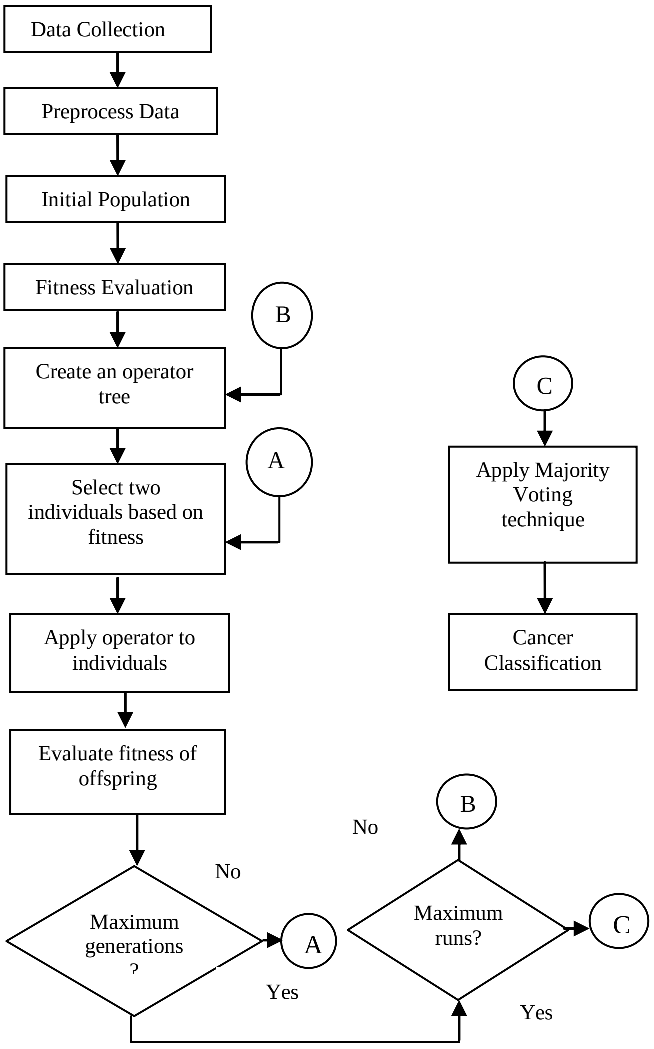

Fig 2: Overall Framework of MVGPC+MGP

Let the S-expression of a rule be represented by R_{expr}, and its output on each sample is a real-valued number. In the typical implementation of a binary GP classifier, the class of a sample Y is predicted as in Eq. (1)

$$Class\ (Y) = \begin{cases} "A", & R_{expr}(Y) \geq 0 \\ "B", & R_{expr}(Y) < 0 \end{cases} \qquad (1)$$

Each S-expression in a population consists of randomly chosen functions and genes. Arithmetic and/or logical functions can be used as functions for the evolution of classification rules. During the coding of GP, choose an appropriate function set depending on the targeted output. Consider a set of functions consisting of either arithmetic or logical functions like {+,-,*,/,sqr,sqrt,exp,and,not,or,>,>=,<,<=,=} or only Boolean functions like {and,or,not,xor,>,>=,<,<=,=} to obtain Boolean outputs(either TRUE or FALSE)

If our targeted output is real, we consider only arithmetic functions like {+,-,*,/,sqr,sqrt,ln,exp,power,sin,cos,tan}.

GP reproduces a population of individuals to solve the problem of classification[6] by executing the following steps:

1.Create initial population of random compositions of functions and terminal sets (genes).

2. Execute each individual in the population on the training samples and assign it a fitness value.

3. While termination criteria are not met, do the following sub steps:

a. Create new offspring by using Meta Genetic Programming by repeatedly applying the following operations to the parents that are selected from the population with a probability based on fitness:

i. Create a new Operator tree by randomly combining different functions.

ii. Apply the individuals selected on the operator tree.

b. Execute each offspring in the new population on the training samples and assign it a fitness value.

```
GenerateRule(Tree t, Integer depth)
If (depth < 1)Then return;
ElseIf (depth = 1) Then
t.value=SelectTerminalRandomly();
t.left=null; t.right=null; return;
Else
node=SelectNodeRandomly();
If (node is a terminal) Then
t.value=node; t.left=null;
t.right=null; return;
ElseIf(node is a unary function) Then
t.value=node; t.right=null;
t.left=new Tree();
GenerateRule(t.left,depth-1);
Else
t.value=node;
t.left=new Tree(); t.right=new Tree();
GenerateRule(t.left,depth-1);
GenerateRule(t.right,depth-1);
```

Fig 3: Pseudo-code for creating a rule in "grow mode"

4.1 Evaluation of Rule

The fitness measure used for classification problems is either the accuracy or the error rate of a predicting program. But, the optimum fitness cannot be achieved through these methods. Matthews[8] proposed a correlation between the prediction and the observed reality as the measure of raw fitness of a predicting program. For a binary classification problem, the Matthews correlation coefficient (MCC) is defined as:

$$MCC = \frac{N_{tp}N_{tn} - N_{fp}N_{fn}}{\sqrt{(N_{tn}+N_{fn})(N_{tn}+N_{fp})(N_{tp}+N_{fn})(N_{tp}+N_{fp})}} \quad (2)$$

where N_{tp}, N_{tn}, N_{fp}, and N_{fn} are the number of true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs), respectively. When the denominator is 0, MCC is set to 0. The standardized fitness of a rule is calculated as shown in Eq.(3)

$$Fitness = \frac{1 + MCC}{2} \quad (3)$$

Since MCC ranges between -1.0 and +1.0, the standardized fitness ranges between 0.0 and +1.0, the higher values being the better and 1.0 being the best. The main objective of GP is to find a rule that can classify all the training samples correctly with the value of fitness as 1.0.

One-versus-rest approach has been considered for the classification of multiclass samples. If there are c classes of samples in a microarray data set, we evolve classification rules in c GP runs. During evolution of a rule i, the samples of class i are treated as positive; other samples as negative and the fitness is calculated. The measures TP, TN, FP, and FN for the fitness of rule i are determined as follows:

- IF $(O(Y) \geq 0)$ AND $(CLASS(Y) = i)$ THEN TP
- IF $(O(Y) < 0)$ AND $(CLASS(Y) \neq i)$ THEN TN
- IF $(O(Y) \geq 0)$ AND $(CLASS(Y) \neq i)$ THEN FP
- IF $(O(Y) < 0)$ AND $(CLASS(Y) = i)$ THEN FN

where O(Y) is the output of the S-expression of rule i on a test sample Y .

The label of a sample is predicted to be the class that has the positive output on the sample. If more than one class has the positive outputs on a sample, the label is determined by randomly picking a class from this set of classes.

4.2 Generation of offspring using Meta Genetic Programming

An operator tree is initially created by combining random functions. The selected parents are then applied to the operator tree to obtain new offsprings. The operator tree's structure decides the output for the operation. This technique must however handle an initial population of operators of unknown quality and incurs an extra computational cost [8]. This technique has a major advantage that it creates more variation in the population. The operator population is treated just as any other individual population and the operators is itself being operated on by other operator population.

4.3 Prediction of Test Class

Majority voting technique is used for predicting test class. The different rules created using MGP runs are evaluated for fitness and the best fit rule is selected to participate in majority voting. The class with the highest vote comes out as the winner class and the test sample is allocated to that sample.

However, for multiclass samples[6], the majority voting technique is applied in a different way. If there are c classes of samples in the microarray data set, we generate a total of v * c rules in v * c GP runs—v rules for each class of samples. During the evolution of a rule for class i, we consider all the samples of class i as positive samples, and the remaining samples as negative samples. Thus, each rule acts as a binary classifier. If the output of the S-expression of a rule for class i has a positive output on a test sample Y, the positive vote in favour of class i is increased by one; otherwise, the negative vote against class i is increased by one. The test sample gets the label of the class that has the highest ratio of positive to negative votes. If two or more ratios are the same, the class is determined by randomly picking one class from the classes corresponding to the ratios. If all ratios are zero, the test sample is treated as misclassified.

5. EXPERIMENTAL RESULTS

5.1 Dataset Description

The data sets included are Leukemia[9], DLBCL[10] and ovarian[11] cancer data sets. Each column contains expression levels of different genes in a single sample, and each row contains expression levels of a single gene in different samples. The relevant genes are selected using Correlation-based feature Selection method with Best-First Search [12].The sample rules produced based on MGP runs for Leukemia cancer data is shown in Table 1.

Table 1. Sample rules evolved in MGP runs

Rules evolved during MGP run
1. (((X2318 - X3144) + (X1313 - X1827)) - SQRT(X1815))
2. (X3019 - SQR(SQRT((X510 / X166))))
(((SQRT(X777) / (X2647 / X2020)) / ((X2636 - X1004) * (X2439 + X1309))) / ((SQRT(X2734) + (X735 * X2006)) + (SQRT(X1412) / SQR(X1249))))
3. (((SQRT(X777) / (X2647 / X2020)) / ((X2636 - X1004) * (X2439 + X1309))) / ((SQRT(X2734) + (X735 * X2006)) + (SQRT(X1412) / SQR(X1249))))
4. (((SQRT(X777) / (X2647 / X2020)) / ((X2636 - X1004) * (X2439 + X1309))) / ((SQRT(X2734) + (X735 * X2006)) + (SQRT(X1412) / SQR(X1249))))
5. (X2735 - SQRT((X1395 + SQR(X3006))))

From the Fig.4, it is clear that MVGPC combined with MGP has lower execution time when compared to pure MVGPC.

Accuracy is calculated based on the number of true positives, true negatives, false positives and false negatives. The sample results for training and test accuracy for MVGPC combined with MGP for Leukemia cancer data when compared to pure MVGPC and SR/SSR are shown in Table 2. From the results, it is clear that MVGPC combined with MGP works better than pure MVGPC or SR/SSR when applied to test samples.

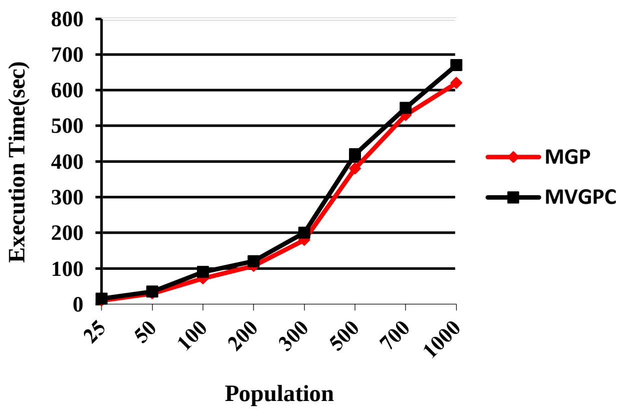

Fig 4: Execution time for MVGPC combined with MGP compared to pure MVGPC

Table 2. Accuracy of MVGPC combined with MGP compared to pure MVGPC and SR/SSR for three cancer datasets

Datasets	Methods	Training Size	Test size	Training accuracy (%)	Test accuracy (%)
Leukemia	MVGPC +MGP	38	34	94.65	92.58
	Pure MVGPC	38	34	89.75	90.16
	SR/SSR	38	34	85.24	84.87
DLBCL	MVGPC +MGP	50	46	93.41	94.23
	Pure MVGPC	50	46	90.78	90.25
	SR/SSR	50	46	89.42	89.85
Ovarian	MVGPC +MGP	30	19	95.27	94.19
	Pure MVGPC	30	19	93.21	92.95
	SR/SSR	30	19	90.35	89.83

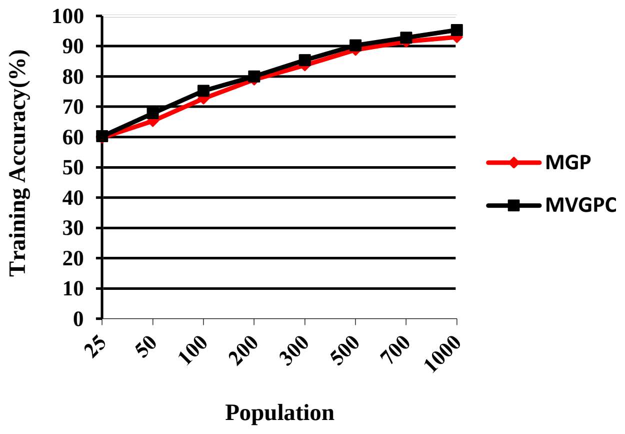

Fig 5: Training accuracy for MVGPC combined with MGP compared to pure MVGPC for Ovarian cancer data

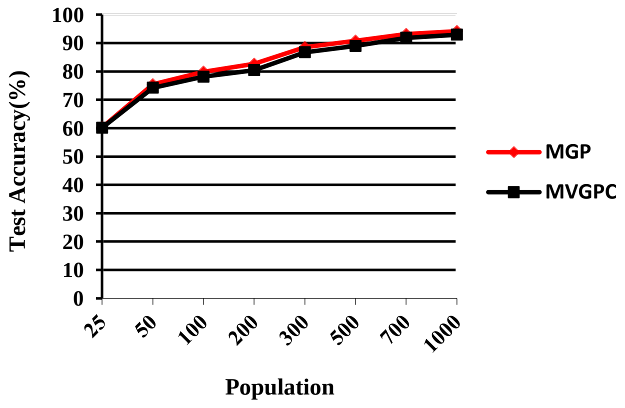

Fig 6: Test accuracy for MVGPC combined with MGP compared to pure MVGPC for Ovarian cancer data

From the above Fig.5 and Fig.6, it is clear that MGP combined with MVGPC works better than pure MVGPC. MGP combined with MVGPC provides an increase of 3-6 % in training accuracy and test accuracy of when compared to pure MVGPC and SR/SSR.

5.2 Feature Ranking

To identify most frequently occurring genes in a particular test sample is of utmost importance in medical field. Using this we can identify the features that are mostly selected during evaluation. The sample result for feature ranking for all cancer data using MGP combined with MVGPC as well as pure MVGPC is given in Table 3. From the result, it becomes clear that MGP combined with MVGPC can identify more genes than pure MVGPC. This enables to identify possible biomarkers for each disease.

Table 3: Feature Ranking for Cancer datasets using MGP combined with MVGPC

Rank	Leukemia		DLBCL		Ovarian	
	Gene ID	*Frequency*	*Gene ID*	*Frequency*	*Gene ID*	*Frequency*
1	X26	45	X1	540	X13	222
2	X11	39	X19	354	X34	176
3	X17	36	X23	196	X8	174
4	X38	29	X26	185	X11	173
5	X6	28	X14	176	X37	170
6	X18	28	X36	172	X52	165

For the given population size, the speed of convergence to the optimum fitness in a run depends on a couple of factors, including the training size and the complexity of the data. For two binary classification problems of same training size and same number of positive and negative samples, MGP may progress to the optimum fitness at different speeds and for two different training sizes, MGP is expected to converge to the optimum in the same average number of generations.

However, on a given data set, it is expected that the average number of generations required by MGP to reach the optimum fitness will increase with the increasing training size. However, due to the huge number of genes and different complexities of training data, MGP may not reach the optimal fitness in every run of an experiment. However, the majority voting technique relies on the number of rules per voting group and on the rate of false prediction (on test samples) by single rules.

6. CONCLUSION

MVGPC combined with MGP will be an effective tool for cancer classification and to identify possible biomarkers for different types of cancers. It provides better accuracy compared to other techniques like pure MVGPC or SR/SSR and also helps to detect more features that are commonly associated with a given type of cancer class. The execution time required is lesser when compared to pure MVGPC. This helps in faster classification of cancer data along with better results. This tool can be used in hospitals and other medical centre to identify cancers in patient samples at an early stage so that patients will have a better chance of survival.

In the future, MGP can be used to effectively analyze and classify different tumors and can be considered as a powerful tool for bioinformatics.

7. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contributions of anonymous referees to this work.

8. REFERENCES

[1] J. R. Koza, "Genetic Programming: On the Programming of Computers by Means of Natural Selection." The MIT Press, 1992.

[2] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov,H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 15, pp. 531–537, 1999.

[3] V. Vapnik, "The Nature of Statistical Learning Theory" NewYork, USA: Springer-Verlag, 1995.

[4] B. Dasarathy, "Nearest Neighbor(NN) Norms: NN Pattern Classification Techniques",IEEE Computer Society Press, 1991.

[5] Bruce Edmonds, "Meta-Genetic Programming: Co-evolving the Operators of Variation", Turkish Journal of Electrical Engineering, Vol.9, November 2001

[6] Topon Kumar Paul and Hitoshi Iba, "Prediction of Cancer Class with Majority Voting Genetic Programming Classifier using Gene Expression Data,"IEEE/ ACM Transactions on Computational Biology and Bioinformatics, Vol. 6, no. 2, pp.353-367, April- June 2009

[7] L. Kuncheva and C. Whitaker, "Measures of diversity in classifier ensembles and their relationships with the ensemble accuracy," Machine Learning, vol. 51, pp. 181–207, 2003.

[8] B. Matthews, "Comparison of the predicted and observed secondarystructure of T4 phage lysozyme," Biochemica et Biophysica Acta., vol.405, pp. 442–451, 1975.

[9] Golub, T. et al. (1999)" Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring" Science, 286, 531–537.

[10] Alizadeh, A. et al. (2000)" Distinct types of diffuse large B-cell Lymphoma Identified by gene expression profiling", Nature, 4051, 503–511.

[11] Ovary Gene, URL: http://public.gnf.org/cancer/ovary/ovary.htm

[12] Cosmin Lazar, Jonatan Taminau, Stijn Meganck, David Steenhoff," A Survey on Filter Techniques for Feature Selection in Gene Expression Microarray Analysis", IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 9, No. 4, July/August 2012