Selecting Informative Genes with Parallel Genetic Algorithms in Tissue Classification

Juan Liu[1,3] **Hitoshi Iba**[2]
liujuan@miv.t.u-tokyo.ac.jp iba@miv.t.u-tokyo.ac.jp
Mitsuru Ishizuka[3]
ishizuka@miv.t.u-tokyo.ac.jp

[1] Department of Computer Science, Wuhan University, Wuhan, 430072,China
[2] Department of Frontier Informatics, University of Tokyo, Hongo 7-3-1 Bunkyo-ku, Tokyo, 113-8656, Japan
[3] Department of Information and Communication Engineering, University of Tokyo, Hongo 7-3-1 Bunkyo-ku, Tokyo, 113-8656, Japan

Abstract

Recent advances in biotechnology offer the ability to measure the levels of expression of thousands of genes in parallel. Analysis of such data can provide understanding and insight into gene function and regulatory mechanisms. Several machine learning approaches have been used to aid to understand the functions of genes. However, these tasks are made more difficult due to the noisy nature of array data and the overwhelming number of gene features. In this paper, we use the parallel genetic algorithm to filter out the informative genes relative to classification. By combing with the classification method proposed by Golub *et al.* [10] and Slonim *et al.* [17], we classify the data sets with tissues of different classes, and the preliminary results are presented in this paper.

Keywords: tissue classification, gene expression level, gene selection, parallel genetic algorithm

1 Introduction

The process of transcribing a gene's DNA sequence into the RNA that serves as a template for protein production is known as gene expression. A gene expression level indicates the approximate number of copies of that gene's RNA produced in a cell; this is thought to correlate with the amount of the corresponding protein made. DNA array technologies have made it straightforward to monitor simultaneously the expression patterns of thousands of genes. These arrays consist of large numbers of specific oligonucleotides or cDNA sequences, each corresponding to a different gene, affixed to a solid surface at very precise locations. When an array chip is hybridized to labeled cDNA derived from a particular tissue of interest, it yields simultaneous measurements of the mRNA levels in the sample for each gene represented on the chip. Since mRNA levels are expected to correlate roughly with the levels of their translation products, the active molecules of interest, array results can be used as a crude approximation to the protein content and thus the 'state' of the sample.

DNA arrays yield a global view of gene expression and can be used in a number of interesting ways, such as clustering the genes into some groups according to the function under certain conditions by unsupervised learning techniques [1, 18, 14, 8, 6, 11, 9] classifying the tissues on the basis of their gene expression patterns by supervised learning techniques [10, 17, 3, 5, 13, 7], and inferring the regulatory network with gene time-course data [15, 19], etc. Recent studies on molecular level classification of tissues have produced remarkable results, and indicated that gene expression assays could significantly aid in the development of efficient cancer diagnosis and classification platforms. However, classification based on the DNA array data is confronted with more challenges. One of the

major challenges is the overwhelming number of genes relative to the number of training samples in the data sets. Many of the genes are not relevant to the distinction between different tissue types and introduce noise in the classification process, and thus potentially drown out the contributions of the relevant ones. Moreover, for diagnostic purposes it is important to find small sets of genes that are sufficiently informative to distinguish between cells of different types. Another challenge is that the data often contain "technical" and "biological" noises. The "technical" noise can be introduced at a number of different stages, such as production of the DNA array, preparation of the samples, hybridization between cDNA and array, and signal analysis and extraction of the hybridization results. The "biological" noise can come from non-uniform genetic background of the samples being compared, or from the impurity or misclassification of tissue samples.

The presented paper goes further in the direction of tissues classification based solely on gene expression levels, and focuses on the topic of selecting a small subset of genes. The selection of informative genes is of fundamental and practical interest. Research in biology and medicine may benefit from the examination of the selected genes to confirm recent discoveries in cancer research or suggest new avenues to be explored. Medical diagnostic tests that measure the abundance of a given protein in serum may be derived from a small subset of informative genes.

Based on whether or not selection is done independently of the classification procedure, gene subset selection algorithms can be classified into two categories. If gene selection is done independent of the classification procedure, the technique is said to follow a filter approach. Otherwise, it is said to follow a wrapper approach [20]. Up to now the presented gene selection techniques belong to the filter approach. While it is generally computationally more efficient than the wrapper approach, the major drawback is that an optimal selection of genes may not be independent of the algorithm to be used to construct the classifier. The wrapper approach, on the other hand, involves the computational overhead of evaluating candidate gene subsets by executing a selected classifying algorithm on the dataset represented using each gene subset under consideration. The genetic algorithm based gene selection method proposed in this paper belongs to this category. Compared to the techniques proposed in the literature, it can handle with multiple selection criteria (e.g., classification accuracy, gene number, etc.) and provide a more robust solution at the expense of increased computation effort.

2 Problem Descriptions and Classical Gene Selection Methods

2.1 Classification Problems

Assume we are given a training set D, consisting of sample and label pairs $< x_i, l_i >$, for $i = 1, 2, \ldots, m$. Each sample x_i is a vector in $\mathbf{R}^N$ that describes expression values of N genes. The label l_i associated with x_i is either 0 or 1 (This paper will limit itself to two-classes classification problems due to the adopted classification method although the GA-based gene selection method can also be easily used in multi-classes classification problems). A classification algorithm is a scalar function f built with D, $f_D{:}x \rightarrow \{0, 1, ?\}$, and new sample x is classified according the value of this function (? represents unclassified). Good classification procedures predict labels that typically match the "true" labels of the samples.

2.2 Ranking Based Gene Selection Methods

A known problem in classification is to find ways to reduce the dimensionality n of the feature space to overcome the risk of "overfitting". It is conceivable to select the best feature subset satisfying a given "model selection" criterion by exhaustive enumeration of all subsets for features. However, exhaustive enumeration is impractical for large numbers of features (in the case of the problem at hand, there are thousands of features/genes) because of the combinatorial explosion of the number of subsets. In order to perform feature selection in large dimensional input spaces, many approaches have been proposed,

almost of which are based on feature-ranking techniques. A fixed number of top ranked features may be selected for further analysis or to design a classifier. Alternatively, a threshold can be set on the ranking criterion. Only the features whose criterion exceeds the threshold are retained. This section simply describes some of the feature-ranking based methods.

2.2.1 Ranking with correlation coefficients

Various correlation coefficients are used as ranking criteria. This paper takes the one by Golub *et al.* and Slonim *et al.* used as an example. It is included in the Golub-Slonim (G-S) classification algorithm [10, 17]. Before performing the classification, genes with the best separation between means for the two classes, as measured by the "G-S correlation" metric are chosen:

$$GS - correlation(g) = (\mu_1^g - \mu_2^g)/(\sigma_1^g + \sigma_2^g),$$

where μ_1^g, σ_1^g and μ_2^g, σ_2^g are the mean and standard deviation for values of gene g among training samples of class 1 and 2, respectively. Genes with the most positive and most negative G-S correlation values are selected in parallel and grouped together in equal number in the final classifier. This method tends to not select genes for which class values have large standard deviations with respect to the training data, though some of those are most relevant and biologically informative.

2.2.2 Ranking with disorder

This method uses only the ranks rather than the actual expression levels to select the genes with the lowest and highest scores in parallel [16]. The score of a gene is defined as the smallest number of swaps of consecutive digits necessary to arrive at a perfect splitting, i.e. the disorder. Formally, let $\vec{g} =< v_1^g, v_2^g, \cdots v_n^g >$ is the vector of the expression levels of gene g in an ascending order, and $\vec{c} =< x_1^g, x_2^g, \cdots x_n^g > (x_i^g \in \{0, 1\})$is the class label vector corresponding to $\vec{g}$ (the label of class 1 is 0, and the label of class 2 is 1), the score of gene g is defined as:

$$Disorder - Score(g) = \sum_{i \in N_2^g} \sum_{j \in N_1^g} h(x_j^g - x_i^g),$$

where N_i^g represents the set of indices belonging to class i and $h(x)$ is the indicator function.

$$h(x) = \begin{cases} 0, \text{ if } x \leq 0, \\ 1, \text{ if } x > 0. \end{cases}$$

The score method measures the correspondence between the expression levels and the class membership. It ignores actual expression values during scoring genes, which would lead to the loss of useful information.

2.2.3 Ranking with likelihood

This method of selecting genes is proposed for a naïve Bayes classifier in [13]. Given the class number c, define $c(c-1)$ different LIK scores:

$$LIK_{j \to k}(g) = \log p(M_j^g | X_j) - \log p(M_k^g | X_j),$$

where X_j are training samples of class j, and $1 \leq j, k \leq c, j \neq k$. M_j^g, M_k^g are the class j, k Gaussian distributions for gene g. We select $c(c-1)$ distinct sets of genes, each maximizing one particular LIK score while merely requiring all others to be greater than zero:

$$GENES_{j \to k}\text{: } LIK_{j \to k}(g) >> 0, \text{ and } LIK_{j' \to k'}(g) > 0, j' \neq k', 1 \leq j', k' \leq c$$

Genes in each $GENES_{j\to k}$ set should therefore best distinguish test samples of class j with respected to the alternative model M_k^g.

In order to compute the LIK scores, it assumes that each class model has equal prior probabilities, class gene data fit a Gaussian distribution, and class attribute values are independent. Obviously, it may not perform in practice as well as expected if some of the assumptions are not justified.

2.2.4 Ranking with $TNoM$ (Threshold Number of Misclassification)

This method is described by Ben-Dor *et al.* [3, 5, 4]. It is based on searching for a decision stump rule. For a given gene, the decision stump rule uses a given expression level to predict the label of an unknown. Formally, a rule is defined by two given parameters d, and t. The predicted class is simply $sign(d(x-t))(d \in \{-1,+1\}$ is the class label parameter; t is the threshold of this gene). The number of errors made by a decision stump rule is defined as:

$$Err(d,t|g) = \sum_i 1\{l_i \neq sign(d(x_i^g - t))\},$$

where x_i^g is the expression value of gene g in the i-th sample, and l_i is the label of this sample. The $TNoM$ score of a gene is simply defined as:

$$TNoM(g) = \min_{d,t} Err(d,t|g)$$

It is the number of errors made by the best stump rule of the gene. The intuition is that this number reflects the quality of decisions made based solely on the expression levels of this gene.

Being aware that the $TNoM$ score just provides partial information about the quality of the predications made by the best rule, Ben-Dor *et al.* discussed another scoring method based on the mutual information between labels and expression values and the derivation of this score. The readers can pursue details in [1].

These feature ranking based gene selection methods select the genes that individually classify best the training data. They eliminate genes that are useless for discrimination, but they do not yield compact gene sets because genes are redundant. Moreover, complementary genes that individually do not separate well the data are missed. Furthermore, the critical weakness of these methods is that they ignore the relationships between genes, being solely based on scoring individual genes. Whereas the features that are top ranked (eliminated last) are not necessarily the ones that are individually most relevant. Only taken together the features of a subset are optimal in some sense. Being aware that a good feature-ranking criterion is not necessarily a good feature subset-ranking criterion. Isabelle *et al.* [12] proposed an iterative procedure called RFE (Recursive Feature Elimination) to select genes subset. However, the RFE is also based on the computing with information about a single gene.

3 Parallel Genetic Algorithms Based Gene Selection Method

3.1 Genetic Algorithms

A genetic algorithm (GA) is a global optimization procedure that uses an analogy of the genetic evolution of biological organisms. It is a heuristic search procedure that modifies function values of individuals coded as binary (or real or symbol) strings, through the application of predefined reproduction operators in a stochastic manner. The string, referred to as a chromosome, is divided into individual sections called genes. The basic GA is shown as Figure 1.

GA has been shown to be a robust and effective search method requiring very little information about the problem to explore a large search space. The only problem dependent part of the GA is the evaluation function. Since many problems can be expressed as optimization problems, they can be

solved by the GA. However, computing cost is the main problem of GA, especially in the situation of large size of population. In order to apply the genetic search into large-scale problems, parallel genetic algorithms (PGA) are widely studied.

3.2 Parallel Genetic Algorithms

There are several ways to parallelize the GA. A very natural way to parallelize GA is to divide the population into subpopulations, run a conventional GA in each subpopulation and allow the periodic communication of information between subpopulations that helps in the search for the solution. The information usually exchanged between subpopulations is a subset of the fittest individuals of each subpopulation. This exchange of individuals is known as migration. This model is called as distributed GA or island model. It is shown as Figure 2.

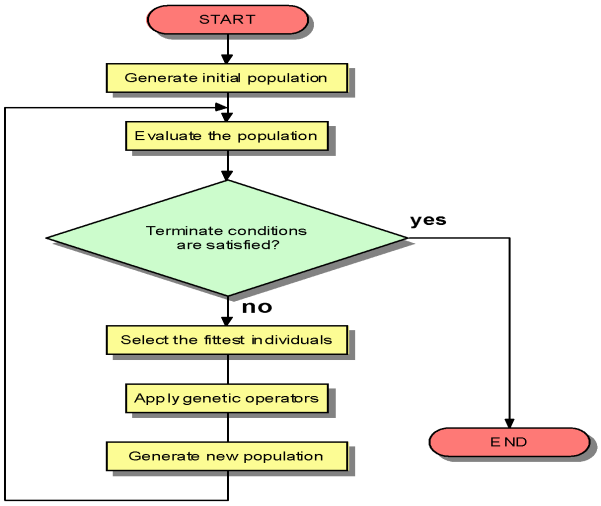

Figure 1: Basic genetic algorithm.

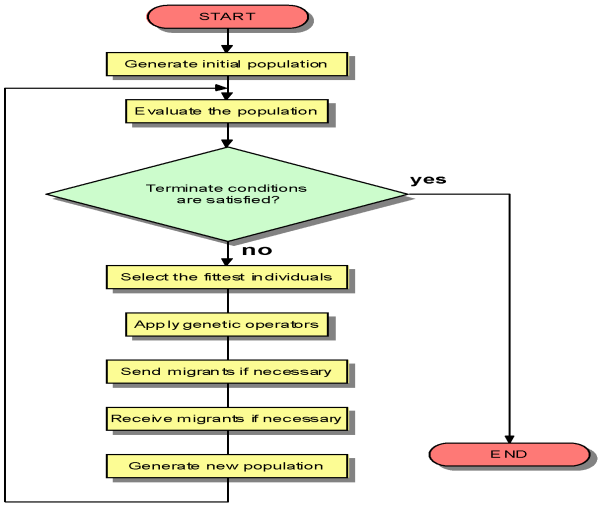

Figure 2: Parallel genetic algorithm.

3.3 PGA Based Gene Selection

In machine learning and pattern recognition areas, there are already some attempts to apply genetic algorithm for the feature (attribute) selection [2, 20]. In the case of gene selection, the genes to be selected can naturally correspond to the features (attributes) in the machine learning areas. Since the problem scale is usually very large due to the large amount of genes, the expensive computing cost of the GA would become intolerable. Wheras the PGA can make use of multiple computing resources at the same time and can divide the larger problem into several smaller ones, It can undoubtedly enhance the efficiency of genetic search. It has also been shown in the evolutionary compution community, that the PGA have higher probability to get the optimal solutions than the GA. Consequently, this paper mainly aims to the PGA based gene selection method. However, we are not plan to compare the performance of PGA with of GA in this paper due to our purpose. The basic idea of PGA based gene selection method is to search the gene subsets with less elements and higher discriminatory power by using PGA as an efficient explore engine of the space of all possible subsets of the whole gene set. Since the PGA search is guided by the fitness values, in order to achieve this goal, the classification performance and the size of the subset should be directly involved in the fitness evaluation. Combined with an efficient classifier, this approach should be feasible.

The population of PGA is divided into subpopulations, and a GA is run on each subpopulation. Each individual in the subpopulation represents a candidate solution to the gene subset selection

problem. There is a very natural way to represent the individuals (and also the gene subsets), namely, a fixed-length binary string representation. In which a bit value of 1 means that the corresponding gene is included in the specified subset, and a value of 0 indicates that the corresponding gene is not included in the subset. The advantage of this representation is that a standard and well understood GA can be used without any modification.

Each member of the current GA population represents a competing gene subset that must be evaluated to provide fitness feedback to the evolutionary process. As mentioned above, the fitness function has to combine two different criteria: the classification accuracy and the size of the subset. The accuracy criterion is achieved by invoking a classifier with the specified gene subset and a set of training data (reduced to include only the gene expression levels of the specified genes). In this paper, a relatively simple form of a 2-criteria fitness function is used:

$$fitness(x) = w_1 * accuracy(x) + w_2 * (1 - dimensionality(x)),$$

where $accuracy(x)$ is the test accuracy of the classifier built with the gene subset represented by x, and $dimensionality(x) \in [0, 1]$ is the dimension of the subset, w_1, w_2 are the weights of $accuracy(x)$ and the gene number beyond x respectively. The PGA is used to maximize the fitness value to find the optimal gene subset. It is obvious that the individual representing the subset with high classification accuracy and low dimension has high fitness value. Of course, this formula is chosen just by experience, and other tries are encouraged.

4 Experiments

4.1 Data Sets

We evaluate the PGA based gene selection method on two data sets described in Table 1.

Table 1: Data sets used in the study.

Data Set	Gene Number	Classes	Train Data	Test Data	Reference
Leukemia	7192	ALL AML	27 11	20 14	*http://www.genome.wi.mit.edu.edu/MPR*
Colon	2000	NORMAL CANCER	22 40	N/A N/A	*http://microarray.princeton.edu/oncology*

Leukemia data set. This data is a collection of 72 expression measurements reported by Golub *et al.* [10] and Slonim *et al.* [17]. It contains a training set composed of 27 samples of acute lymphoblastic leukemia (ALL) and 11 samples of acute myeloblastic leukemia (AML), and an independent test set composed of 20 ALL and 14 AML samples. High-density oligonucleotide microarrays containing 7129 probes for 6817 human genes were used. The data is available at *http://www.genome.wi.mit.edu.edu/MPR.*

Colon data set. This data set is a collection of 62 expression measurements from colon biopsy samples reported by Alon *et al.* [1]. It contains 22 normal and 40 colon cancer tissues. 2000 genes with highest minimal intensity across 62 tissues are used. The colon data can be found at the web site *http://microarray.princeton.edu/oncology.*

4.2 Classifier

We use the classifier proposed by Golub *et al.* [10] and Slonim *et al.* [17] to evaluate the gene selecting method:

$$class(x) = sign\{ \sum_{\text{gene } g \text{ in classifier}} [(\mu_1^g - \mu_2^g)/(\sigma_1^g + \sigma_2^g)][x_g - (\mu_1^g + \mu_2^g)/2]\},$$

where μ_1^g, σ_1^g and μ_2^g, σ_2^g are the mean and standard deviation for values of gene g in the classifier of class 1 and 2, respectively, and x_g is the expression values of gene g in sample x. if the computed value is positive, sample x belongs to class 1, negative value means x belongs to class 2. This classifier is only applicable to data sets with two classes.

4.3 Methods and Results

In order to test the PGA-based method, we set up the experiments shown as Figure 3. For the PGA module, the parameters are shown as Table 2 and Table 3. Being aware that classification accuracy is still much more important than the gene number though the aim of this paper is to reduce the number of genes, we set $w_1 = 0.75$, and $w_2 = 0.25$ to evaluate the fitness values to emphasize the effect of the accuracy.

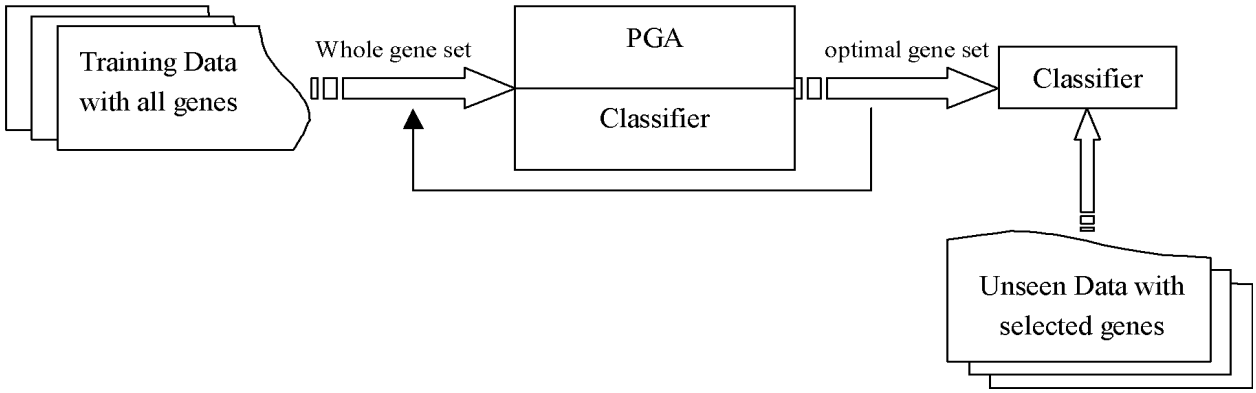

Figure 3: Setup for experiments of PGA-based method.

Table 2: Parameters of the communication.

Processors	Migrants	Migration rate	Topology
2	20	0.002	Bi-directional rings

Table 3: Parameters of the GA in each processor.

Population	Trials	Crossover Rate	Mutation Rate	Replacement Ratio	Select Strategy
1000	400000	0.6	0.001	1.0	Elitist

The initial training samples are sent to the PGA-Classifier component for gene selection. The hybrid component runs 10 times and the best subset of the 10 runs is selected as the final optimal subset, which is used to construct the classifier to classify the unseen samples.

As for the PGA-Classifier component, the training data is randomly shuffled and generate two subsets, one subset containing 50% of the samples for constructing the classifier and the other subset containing 50% of the samples for evaluating the classifier and then calculate the fitness values of individuals.

After 10 runs of PGA-Classifier component, we test the selected gene subset on the data sets metioned above, and follow the same lines as [10, 17, 13] to show whether our results outperform them. Concretely, in the case of leukemia data set, we use the same assessing procedure as in [10, 17], i.e., two-step procedure. The accuracy of the classifier is first tested by LOOCV (leave one out cross validation) method, in which one sample in the training set is withheld, the remaining samples of the training set are used to build a classifier to predicate the class of the withheld sample, and the cumulative error rate is calculated. Then a final predicator is built based on the whole training set and its accuracy on the test set of samples is accessed. In the case of colon data set, we only use the LOOCV method to test the classifier just same as [13]. The results on the two data sets are shown as Table 4.

Table 4: Results of experiments.

Data set	#Genes	Acc. LOOCV	Acc. Test	Exp. No.	Gen No.	Trial No.
Leukemia	29	0.95	0.88	5	32	32284
Colon	30	0.92	N/A	6	354	301892

For leukemia data set, a subset with 29 genes is found in the 32^{th} generation, 32284^{th} fitness evaluation in the 5^{th} experiment. The 29-gene classifier predicates 36 of 38 samples in the training set and 30 of 34 samples in the testing set correctly though in which 3 control genes meaningless to classification are included. Whereas, about 50 genes are required to achieve the approximate result in the original work of Golub *et al.* [10] and Slonim *et al.* [17]. When dropping the 3 genes from 29-gene set, even more higher accuracy on the test data set(0.91) is achieved. For colon data set, a subset with 30 genes is found in the 354^{th} generation, 301892^{th} fitness evaluation in the 6^{th} experiment, and the LOOCV accuracy of 30-gene classifier is 0.92. The result also outperforms some other algorithms such as G-S and NB [13], in which the LOOCV accuracies are below 0.90 with gene numbers varying from 10 to 500. The PGA's average performance of 10 experiments is shown as Figure 4 and Figure 5.

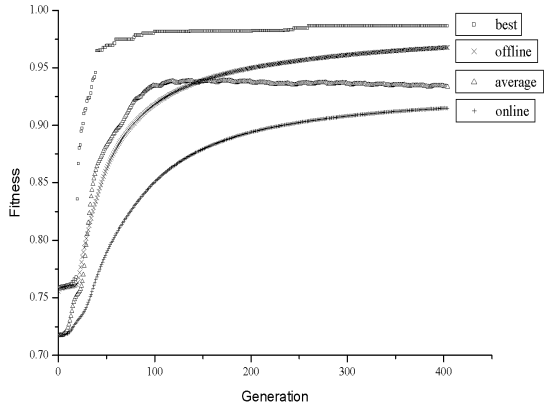

Figure 4: Performance for leukemia data set.

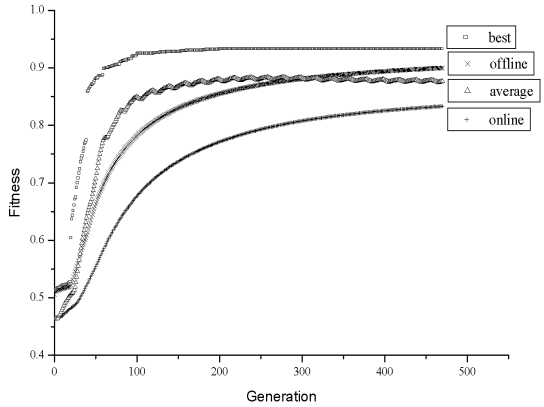

Figure 5: Performance for colon data set.

The selected gene subset (3 control genes excluded) for Leukemia is shown as Figure 6, where the red boxes represent positive values, green boxes represent negative values, and black boxes represent zeroes. By using CLUSTER software of Eisen's Lab, genes are clustered into 2 categories: one with 9 genes which seem informative to AML tissues and the other one with 17 genes. The top 8 of the 17 genes seem more informative to ALL tissues than other genes.

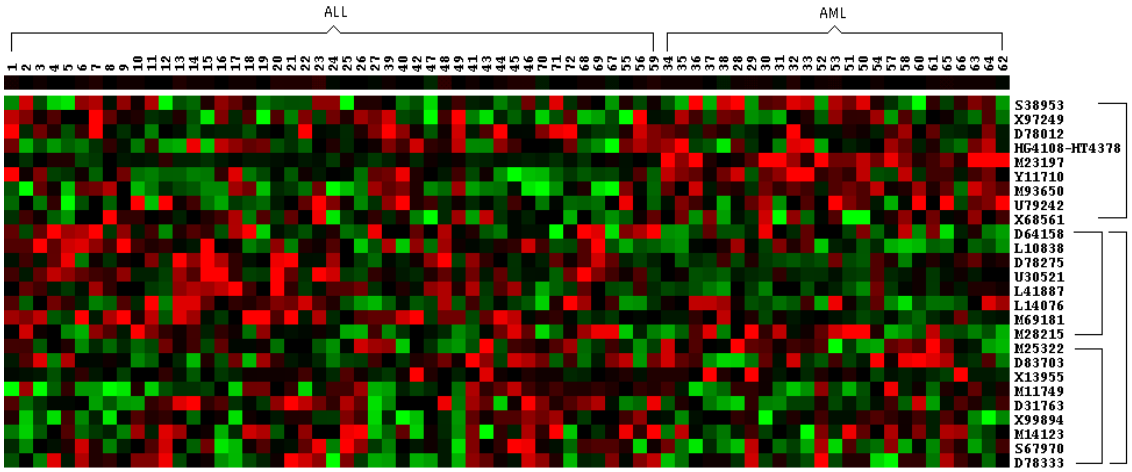

Figure 6: Selected gene subset of Leukemia-2 data.

5 Conclusions and Future Works

This paper addresses the question of gene selection in tissue classification based on expression data. Classical ranking based methods select the top ranked genes, and are usually efficient. However, the set of top ranked genes may not be the optimal gene subset for tissue classification. Whereas GA based method directly select the optimal gene subset, and can provide a more robust solution at the expense of increased computation effort. PGA model used in the paper divided the population into subpopulations on which a GA is run, the search task of optimal gene subset then can be distributed on several CPUs/computers, which can preserve the diversity of population as well as enhance the efficiency of the algorithm. Two open data sets are used to test the PGA based gene selection method, and the experiment results show that it can get more compact gene subset.

However, as mentioned above, the final subset PGA found may not the most compact one due to the problem itself or the fitness function. Since the fitness function used in this paper seems too simple to guide the PGA to perform effective search, a more sophisticated formula such as one useing an accuracy normalized by class population size instead of accuracy will be powerful.

In the present work, the classifier Golub *et al.* proposed is used as an evaluation of the individuals, which leads to it can only deal with 2-class problem. Future work will experiment with the extension of the method to multi-class tissue classification problems. Some other future topics include: extensive experimental comparison of the performance of the proposed approach with that of conventional methods, more principled design of fitness function using domain knowledge as well as mathematically well-founded tools of multi-attribute utility theory, and the design of new domain-related genetic operators.

References

[1] Alon, U. *et al.*, Broad patterns of gene expression revealed by clustering analysis of tumor colon tissues probed by oligonucleotide arrays, *PNAS*, 96:6745–6750, 1999

[2] Bala, J., Huang, J., Vafaie, H., and Wechsler, H., Hybrid learning using genetic algorithms and decision trees for pattern classification, *Proc. 14th IJCAI Conference*, Montreal, August 19–25, 1995.

[3] Ben-Dor, A. *et al.*, Tissue classification with gene expression profiles, *J. Computational Biology*, 7(3/4):559–583, 2000.

[4] Ben-Dor, A. *et al.*, Scoring genes for relevance, *Technical Report 2000-38*, School of Computer Science & Engineering, Hebrew University, Jerusalem, 2000.

[5] Ben-Dor, A., Friedman, N., and Yakini, Z., Class discovery in gene expression data, *Proc. of the 5th Annual International Conference on Computational Molecular Biology*, 31–38, 2001.

[6] Ben-Dor, A., Shamir, R., and Yakini, Z., Clustering gene expression patterns, *J. Computational Biology*, 6(3/4):281–297, 1999

[7] Brown, M.P.S. *et al.*, Knowledge-based analysis of microarry gene expression data by using support vector machines, *PNAS*, 97(1):262–267, 2000.

[8] Caron, H. *et al.*, The human transcriptome map: clustering of highly expressed genes in chromosomal domains, *Science*, 291:1289–1292, 2001.

[9] Eisen, M.B. *et al.*, Cluster analysis and display of genome-wide expression patterns, *PNAS*, 95:14863–14868, 1998.

[10] Golub, T.R. *et al.*, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science*, 286(15):531–537, 1999.

[11] Heyer, L.J., Kruglyak, S., and Yooseph, S., Exploring expression data: identification and analysis of coexpressed genes, *Genome Research*, 9:1106–1115, 1999.

[12] Isablle, G. *et al.*, Gene selection for cancer classification using support vector machines, 2001. http://citeseer.nj.nec.com/402272.html

[13] Keller, A.D., Schummer, M., *et al.*, Bayesian classification of DNA array expression data, *Technical Report UW-CSE-2000-08-01*, Department of Computer Science & Engineering, University of Washington,Seattle, 2000.

[14] Kim, J.H., Ohno-Machado, L., Kohane, I.S., Unsupervised learning from complex data: the matrix incision tree algorithm, *PSB2001*, 6:30–41, 2001.

[15] Maki, Y., Tominaga, D., Okamoto, M., Watanabe, S., and Eguchi, Y., Development of a system for the inference of large scale genetic networks, *PSB2001*, 6:446–458, 2001.

[16] Park, P.J., Pagano, M., and Bonetti, M., A nonparametric scoring algorithm for identifying informative genes from microarry data, *PSB2001*, 6:52–63, 2001.

[17] Slonim, D.K. *et al.*, Class predication and discovery using expression data, *Proc. of the 4th Annual International Conference on Computational Molecular Biology*, 263–272, 2000.

[18] Tamayo, P. *et al.*, Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differnetiation, *PNAS*, 96:2907–2912, 1999.

[19] Weaver, D.C., Workman, C.T., and Stormo, G.D., Modeling regulatory networks with weight matrics, *PSB1999*, 4:112–123, 1999.

[20] Yang, J. and Honavar, V., Feature subset selection using a genetic algorithm, 1997. http://citeseer.nj.nec.com/yang97feature.html