Bio-Medical Materials and Engineering 26 (2015) S1993–S2002
DOI 10.3233/BME-151503
IOS Press

A genetic filter for cancer classification on gene expression data

Yong-Hyuk Kim [a] and Yourim Yoon [b,*]

[a] *Department of Computer Science & Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 139-701, Republic of Korea*

[b] *Department of Computer Engineering, College of Information Technology, Gachon University, 1342 Seongnam-daero, Sujeong-gu, Seongnam-si, Gyeonggi-do 461-701, Republic of Korea*

Abstract. We present a new genetic filter to identify a predictive gene subset for cancer-type classification on gene expression profiles. This approach pursues to not only maximize correlation between selected genes and cancer types but also minimize inter-correlation among selected genes. The proposed genetic filter was tested on well-known leukemia datasets, and significant improvement over previous work was obtained.

Keywords: Gene selection, filter method, genetic algorithm, cancer classification, gene expression data

1. Introduction

As the technology of microarray has grown, researchers have gotten to investigate many gene expression patterns simultaneously. Microarray has been an experimental tool to extract functional information from genome [1,2]. The disease-type classification is representative in microarray applications. In recent years, ones used microarray to profile the gene expression pattern of abnormal or normal cell in tumor, e.g., leukemia [3]. Such study sheds light on obtaining bio-markers for classifying cancers. Clustering analysis [4–8] is prevalent for the analysis of microarray data. Some studies on clustering analysis have focused on biclustering of gene expression data [9–11]. The clustering analysis clusters genes that have closely related expression patterns which enable us to get some insights into gene function and interaction between genes.

Microarray has been extensively adopted to profile gene expression data of tumors and applied to cancer classification, but its success largely depends on the tools of data mining. Because, among too many gene expression data, only a part give distinct expression levels for different disease types. Hence it is quite important to use the tools that can identify informative genes from a large number of genes polluted with noise.

[*] Address for correspondence: Yourim Yoon, Department of Computer Engineering, College of Information Technology, Gachon University, Seongnam-daero, Sujeong-gu, Seongnam-si, Gyeonggi-do 461-701, Republic of Korea. Tel.: +82.31.750.5326; Fax: +82.31.750.5662; E-mail: yryoon@gachon.ac.kr.

Microarray data make up of many gene expression data and relatively small samples. To overcome this unbalance, we select an informative gene subset. Our study is to propose a filter method for the selection based on a genetic algorithm.

There may exist a number of gene subsets that can differentiate between disease types of samples. The proposed approach is to get various such subsets for classification and then estimate the gene importance from correlation between each pair of genes in the subset. In the case that selected gene subset was used for classification on test data, samples could be well classified. Other methods that identify a gene subset for classification have also been presented [3,12–15]. We investigate the patterns of identified genes and examine the classification reliability of the identified genes from test data. We also do the sensitivity analysis of the test results on selected genes. We divide the dataset into training and test samples differently. Training and test sets are used to identify and evaluate a subset of predictive genes, respectively.

In this study, we considerably extend our preliminary work [16]. Using leukemia dataset as a benchmark dataset, we give an analysis on the leukemia dataset using a genetic filter to identify a subset of predictive genes that can distinguish between the two disease types: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). We compare the results with those of previous work. The main difference with our preliminary work [16] lies in that: (i) we improves the genetic parameters including evaluation function, (ii) we provides the detailed analysis of the proposed genetic filter through considerably extended experiments, which were conducted on varying parameters, and (iii) we reports statistically significant results using bootstrapping, which is a re-sampling technique for estimating summary statistics.

The remainder of this paper proceeds as follows. We introduce previous work including cancer-type predictor and leukemia dataset in Section 2. In Section 3, we present a genetic filter for gene identification. We provide empirical analysis in Section 4. In Section 5, we draw conclusions.

2. Previous work

Golub et al. [3] presented an effective method to identify a predictive gene subset for cancer classification. They used a neighborhood analysis in selecting a gene subset that can distinguish between the two cancer types: AML and ALL, based on a separation measure similar to t-statistic. The subset consisting of fifty genes best discriminating between AML and ALL in training dataset was selected as a cancer-type predictor. It correctly classified 36 of the 38 training samples. This gene subset was subsequently used to predict the cancer type on test dataset. In their experiments, 29 of the 34 test samples were classified correctly, where 4 of the 5 ($= 34 - 29$) samples were not classified, i.e., undecided and the remaining one sample was misclassified, i.e., error.

Golub et al. [3] also proposed a predictor that uses a predictive gene subset with a fixed size and predicts a test sample with the expression levels of these predictive genes. The pseudo-code of the cancer-type predictor is given in Figure 1. Each predictive gene takes a vote on the cancer type, with the weight of each vote depending on the expression levels in the test sample and the correlation between that gene and the cancer-type distinction. The weights of the votes are summed to decide the winning cancer type and a real-valued prediction strength ranging between -1 and 1. The test sample is classified into the winner cancer type when the prediction strength is larger than a given threshold, and it is regarded undecided in other cases.

The leukemia dataset contains 6,817 gene expression levels in 72 samples, among which 25 samples were classified as AML and the remaining 47 ones as ALL [3]. Each sample has the gene expression

```
CancerTypePredictor(sample $x = (x_1, x_2, \ldots, x_k)$)
{
    // $x_i$ is the $i$-th gene expression level for sample $x$, and $k$ is the size of the identified gene subset
    $V_m \leftarrow 0, V_l \leftarrow 0$;
    for each gene $i$,
        $(\mu_m(i), \mu_l(i)) \leftarrow$ averages of the $i$-th gene expression levels for the samples in AML and ALL, respectively;
        $(\sigma_m(i), \sigma_l(i)) \leftarrow$ standard deviations of the $i$-th gene expression levels for the samples in AML and ALL, respectively;
        $\rho'(i, C) \leftarrow (\mu_m(i) - \mu_l(i))/(\sigma_m(i) + \sigma_l(i))$;
        $v_i \leftarrow \rho'(i, C) \cdot (x_i - (\mu_m(i) + \mu_l(i))/2)$;
        if $v_i > 0$ then $V_l \leftarrow V_m + v_i$;
        else $V_l \leftarrow V_l - v_i$;
    prediction strength $\leftarrow (V_m - V_l)/(V_m + V_l)$;
    if |prediction strength| $<$ threshold then return undecided;
    else if prediction strength $\geq$ threshold then return AML;
    else return ALL; // prediction strength $\leq -$threshold
}
```

Note: the threshold of 0.3 is used following [3].

Fig. 1. Pseudo-code of cancer-type predictor [3,16].

levels in microarray data. The dataset has been widely used in other studies [17–22]. We divided the dataset into 38 training samples and 34 test samples as in [3]. The training samples were used to select a gene subset that can distinguish between the two cancer types: AML and ALL. The fifty most predictive genes identified by the training samples were validated, to classify the test samples.

3. A genetic filter

Genetic algorithms (GAs) have been popular for feature selection [23–25]. We propose a new GA to identify a predictive gene subset. It chooses genes using the training samples. Usually a GA for feature selection is used as a wrapper method, which maximizes the accuracy on the training dataset. But the proposed GA is not a wrapper method but a filter method. One of the notable features is that the proposed GA searches for a predictive gene subset based on correlation-based evaluation. The solution set has exponentially many elements. When the number of genes to select is given, the best subset of size k can be obtained by considering all possible cases, i.e., $\binom{n}{k}$ cases. The proposed GA alternatively searches the solution set to get a good predictive gene subset with a given fixed size.

The dataset is divided into the training and test sets. The proposed GA chooses a given number of genes by conducting search on the training samples. After the proposed GA chooses a predictive gene subset, the predictor using the subset runs on the test samples.

We use the general structure of a steady-state GA [26]. An individual is encoded by binary string, in which a gene has value one when belonging to the predictive gene subset; in the other case, it has value zero. First, the proposed GA makes P initial subsets randomly. The only constraint on an individual is that the number of one's is fixed. The population size P is set to be 100. A fitness computed from objective value is assigned to each individual in the population. The proportional selection scheme is used. Crossover operator makes an offspring through recombining parts of both parents. One-point crossover, which is the most popular, is used. As mutation, swap mutation, which exchanges the values of a pair of genes chosen randomly, is used. An offspring obtained after crossover and mutation operators may not be feasible. That is, it may not meet the constraint with the fixed number of genes to be selected. The

GA then randomly chooses genes on the individual and alters the required number of zero's to one's (or one's to zero's). This repair process can also give some effect of mutation. After generating a feasible offspring, an individual in the population is replaced with the offspring. The replacement of [26] is used. First, the closer parent with respect to Hamming distance tries to be replaced with the offspring. If the closer parent is better than the offspring, the remaining parent tries to be replaced with the offspring. This replacement with parents is done only if one of both parents is worse than the offspring. If both parents are better than the offspring, the worst individual in the population is replaced with the offspring. As termination condition, the number of consecutive fails in parent replacement is used. The number was set to be twenty.

Our evaluation is to obtain a predictive gene subset, highly related to the cancer type and low related to other genes in the predictive gene subset. The proposed GA minimizes the objective $p \cdot \overline{|\rho(X,Y)|} + |\rho'(X,C)|^{-1}$, in which p is the inter-correlation weight factor, $\rho(x,y)$ means the correlation value between genes x and y, and $\rho'(x,C)$ is the correlation value between gene x and the cancer-type, which is explained in [3] and given in Figure 1. Inter-correlation weight factor p is used in the fitness evaluation. If p is zero, we just finds a gene subset highly related to the cancer-type. When p is larger than zero, we prefer a gene subset low related to other genes in the subset. It has an effect of making the selected genes be uniformly distributed in the gene space. In the case that p is less than zero, a gene subset closely related to other genes in the subset has high fitness. It makes the selected genes be clustered in the gene space.

4. Experimental results

Let k be the number of predictive genes to select. The predictive gene subset identified by Golub et al. [3] makes up of the $k/2$ genes that are the most closely related to the cancer-type AML and the $k/2$ genes that are the most closely related to the cancer-type ALL. In other words, at correlation ρ' in Figure 1, the topmost $k/2$ genes and bottommost $k/2$ ones are selected. "Greedy" selects the topmost k genes at the absolute value of ρ'. "Random" randomly selects k genes among a given candidate gene set. "GA" optimally selects k genes among a given candidate gene set using the proposed genetic filter.

It is important to correctly classify as many samples as possible. It is much more crucial to lower misclassified (error) samples [3], rather than to lower undecided samples, To validate a predictive gene subset, we conducted experiments with two validation procedures. We go with the validation procedure of [3]. It proceeds the following two steps. (i) First, the accuracy is validated by leave-one-out-cross-validation (LOOCV) on the training set. We withhold a sample in the training set, build a predictor using the remaining training samples, and predict the cancer type of the withheld sample. This process is performed for each training sample, and we get the cumulative error. (ii) Then we build a final predictor using all the training samples, and then evaluate its accuracy on the test samples.

The prediction is also repeated for many different bootstrap samples. Bootstrapping introduced by Efron [27,28] is to estimate the generalization of a predictor based on re-sampling. We draw the $(n - v)$ samples without repetition for dataset. in which n and v mean the numbers of samples in the universe and test datasets, respectively. The training and test sets are disjoint for each bootstrap sample. The predictor is trained on training samples and its accuracy is obtained from the prediction on test samples. Following a given size of divided samples, we set n and v to be 72 and 34, respectively.

The average sizes of candidate gene sets restricted with correlation ρ' are given in Table 1. Tables 2-4 show the experimental results with 50, 20, and 100 as the number of genes to select, respectively. Note

Table 1
Number of genes with constraints w.r.t. correlation ρ'.

	#genes		
$\|\rho'\|$	Ave	Min	Max
Not restricted (i.e., all genes)	6817	6817	6817
Larger than 0.1	5158.35	4584	5739
Larger than 0.3	2105.42	1440	3173
Larger than 0.5	663.94	358	1378
Larger than 0.7	177.85	70	474
Larger than 0.8	91.50	32	255

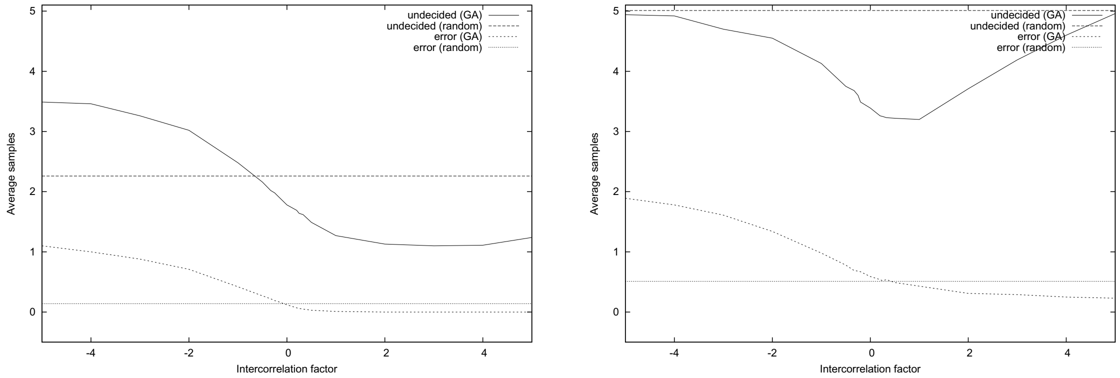
Table 2
Results with 50 selected genes, averaged over 1,000 bootstrap samples.

	Training data		Test data	
	Undecided	Error	Undecided	Error
Method	Ave(SD)	Ave(SD)	Ave(SD)	Ave(SD)
Random (all genes)	17.52(3.69)	1.64(1.25)	15.32(3.47)	1.59(1.62)
Random ($\|\rho'\| > 0.1$)	15.32(3.94)	1.23(1.17)	15.16(3.61)	1.49(1.64)
Random ($\|\rho'\| > 0.3$)	8.64(2.24)	0.52(0.70)	11.88(3.12)	1.20(1.48)
Random ($\|\rho'\| > 0.5$)	4.33(1.83)	0.25(0.45)	8.22(2.89)	0.73(1.15)
Random ($\|\rho'\| > 0.7$)	2.26(1.31)	0.14(0.35)	5.01(2.60)	0.51(0.84)
Golub et al. [3]	1.82(1.08)	0.05(0.21)	2.95(1.56)	0.47(0.59)
Greedy	1.84(1.12)	0.12(0.32)	3.19(1.69)	0.64(0.76)
GA (all genes)	1.60(1.11)	0.03(0.17)	4.41(2.21)	0.44(0.62)
GA ($\|\rho'\| > 0.1$)	1.49(1.09)	0.01(0.12)	4.43(2.05)	0.39(0.59)
GA ($\|\rho'\| > 0.3$)	1.16(0.96)	0.00(0.03)	3.95(2.02)	0.39(0.56)
GA ($\|\rho'\| > 0.5$)	0.98(0.87)	0.00(0.00)	3.81(1.96)	0.35(0.54)
GA ($\|\rho'\| > 0.7$)	1.13(0.93)	0.00(0.00)	3.71(1.96)	0.31(0.52)

Note: the numbers of training and test samples are 38 and 34, respectively.
Inter-correlation weight factor $p = 2$ in GAs.

that previous work [3,16] used only one number, 50, as the size of gene subset. The results are averaged over 1,000 bootstrap samples. The more candidate genes were considered (i.e., lower $|\rho'|$), the more errors were obtained. The correlation ρ' is shown to be suitable for assessing the predictivity of genes. But, Greedy that selects genes only with the topmost $|\rho'|$ dominated by Golub et al. [3] and GA. It hints that an additional measure is necessary for finding a more predictive gene subset. In the case of GA also considering inter-correlation as a lowering factor, it performed the best. GA had the best accuracy on training set, and it showed nearly zero error in LOOCV on the set. Undecided samples in test set were less in Greedy and Golub et al. [3] than in GA. However GA reported the lowest misclassified (error) rate on test set.

We used inter-correlation weight factor $p = 2$ as our default setting. This weight factor largely affected the performance as shown in Figures 2 and 3, where the number of selected genes is 50. Larger positive factors led to lower errors but much more undecided samples in test data. In the case of negative inter-correlation weight factor ($p < 0$) which selects a subset of genes closely related to other genes in the



(a) Training data

(b) Test data

Fig. 2. Results of GA ($|\rho'| > 0.7$) according to inter-correlation weight factor p.

Table 3

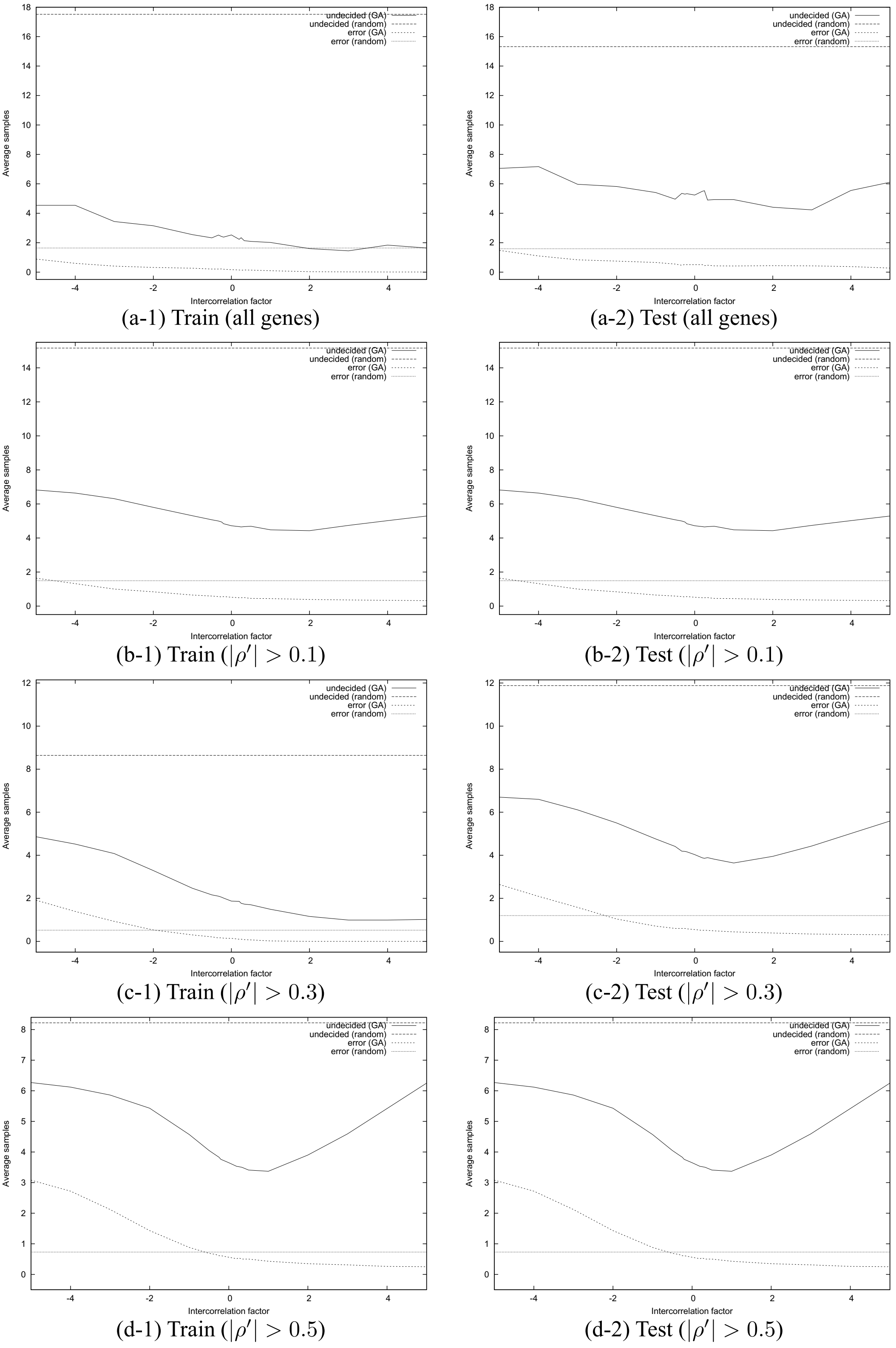
Results with 20 selected genes, averaged over 1,000 bootstrap samples.

	Training data		Test data	
	Undecided	Error	Undecided	Error
Method	Ave(SD)	Ave(SD)	Ave(SD)	Ave(SD)
Random (all genes)	18.54(4.15)	5.60(1.92)	15.63(3.86)	4.27(2.31)
Random ($|\rho'| > 0.1$)	14.26(3.72)	2.50(1.67)	14.06(3.63)	2.95(2.14)
Random ($|\rho'| > 0.3$)	9.55(2.73)	1.02(1.04)	11.50(3.19)	2.09(1.91)
Random ($|\rho'| > 0.5$)	5.39(2.04)	0.46(0.67)	8.71(2.92)	1.21(1.48)
Random ($|\rho'| > 0.7$)	2.90(1.51)	0.24(0.47)	5.68(2.66)	0.83(1.09)
Random ($|\rho'| > 0.8$)	2.37(1.31)	0.22(0.44)	4.33(2.28)	0.75(0.93)
Golub et al. [3]	1.96(1.15)	0.09(0.29)	2.84(1.50)	0.62(0.65)
Greedy	1.93(1.18)	0.19(0.40)	2.92(1.60)	0.92(0.94)
GA (all genes)	1.75(1.23)	0.04(0.18)	5.15(2.34)	0.56(0.80)
GA ($|\rho'| > 0.1$)	1.64(1.15)	0.02(0.14)	4.33(2.10)	0.57(0.70)
GA ($|\rho'| > 0.3$)	1.33(1.07)	0.01(0.11)	4.06(2.08)	0.63(0.76)
GA ($|\rho'| > 0.5$)	1.23(1.04)	0.00(0.04)	4.19(2.14)	0.56(0.71)
GA ($|\rho'| > 0.7$)	1.12(0.97)	0.00(0.05)	4.23(2.08)	0.55(0.67)
GA ($|\rho'| > 0.8$)	1.26(1.01)	0.00(0.05)	3.95(1.96)	0.53(0.66)

Note: the numbers of training and test samples are 38 and 34, respectively.
Inter-correlation weight factor $p = 2$ in GAs.

subset, because identified genes are highly biased, its error was much higher than that of positive inter-correlation weight factor ($p > 0$).

We also examined the performance according to the number of predictive genes to select. Figure 4 shows the results for the four methods: Golub et al. [3], Greedy, Random, and GA. The number varies from 10 to 300. In this experiment, Random and GA considered the candidate gene set with $|\rho'| > 0.5$. When the number is less than 50, more genes showed better performance in test data. But when the number is greater than 100, larger number led to slightly lower errors but much more undecided samples in test data.



(a-1) Train (all genes)

(a-2) Test (all genes)

(b-1) Train ($|\rho'| > 0.1$)

(b-2) Test ($|\rho'| > 0.1$)

(c-1) Train ($|\rho'| > 0.3$)

(c-2) Test ($|\rho'| > 0.3$)

(d-1) Train ($|\rho'| > 0.5$)

(d-2) Test ($|\rho'| > 0.5$)

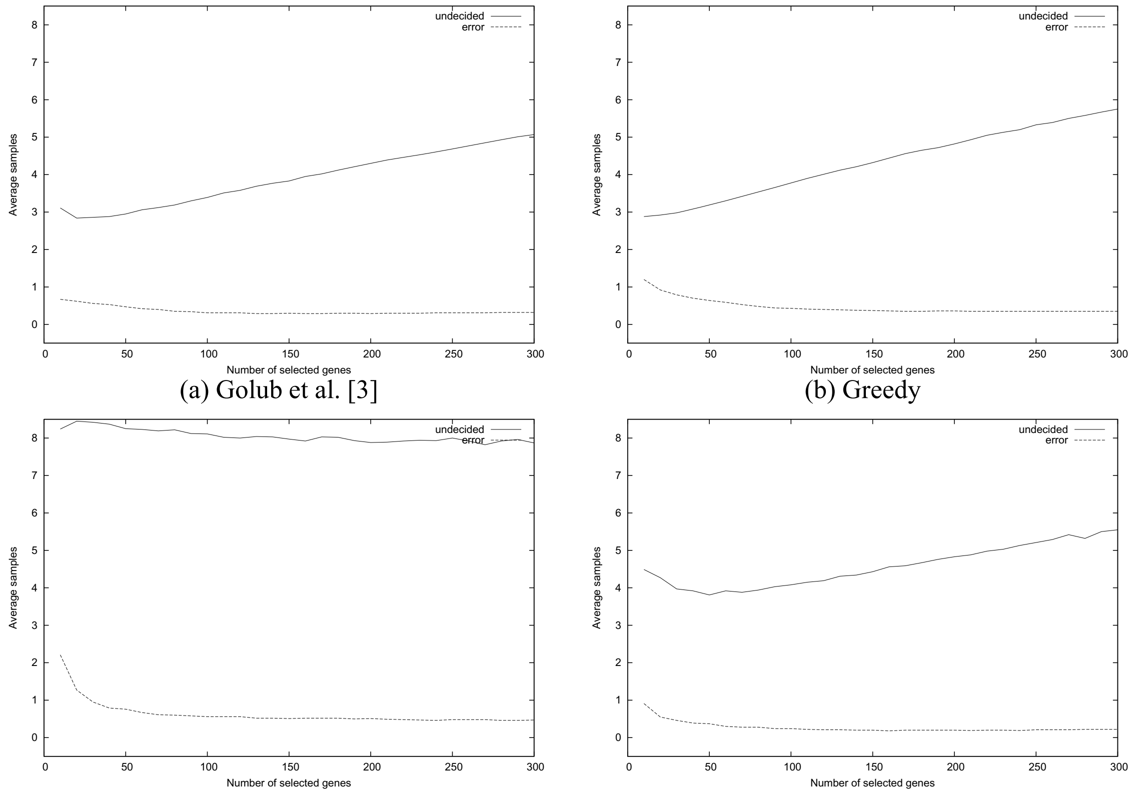
Fig. 3. Results of GAs according to inter-correlation weight factor p.

Table 4

Results with 100 selected genes, averaged over 1,000 bootstrap samples.

	Training data		Test data	
	Undecided	Error	Undecided	Error
Method	Ave(SD)	Ave(SD)	Ave(SD)	Ave(SD)
Random (all genes)	16.28(3.74)	0.92(0.94)	15.04(3.23)	0.95(1.31)
Random ($\lvert\rho'\rvert > 0.1$)	15.05(3.60)	0.85(0.91)	15.57(3.51)	1.08(1.40)
Random ($\lvert\rho'\rvert > 0.3$)	8.34(2.21)	0.41(0.61)	12.03(3.14)	0.89(1.38)
Random ($\lvert\rho'\rvert > 0.5$)	3.82(1.78)	0.20(0.41)	8.16(2.88)	0.57(1.04)
Golub et al. [3]	1.68(1.02)	0.04(0.21)	3.39(1.85)	0.31(0.50)
Greedy	1.76(1.06)	0.09(0.29)	3.78(2.02)	0.43(0.65)
GA (all genes)	2.19(1.29)	0.09(0.29)	5.85(2.45)	0.41(0.61)
GA ($\lvert\rho'\rvert > 0.1$)	1.84(1.14)	0.04(0.21)	5.46(2.28)	0.31(0.53)
GA ($\lvert\rho'\rvert > 0.3$)	1.35(0.94)	0.01(0.07)	4.56(2.05)	0.27(0.49)
GA ($\lvert\rho'\rvert > 0.5$)	1.09(0.93)	0.00(0.00)	4.12(2.04)	0.23(0.47)

Note: the numbers of training and test samples are 38 and 34, respectively.
Inter-correlation weight factor $p = 2$ in GAs.



(a) Golub et al. [3]

(b) Greedy

(c) Random ($|\rho'| > 0.5$)

(d) GA ($|\rho'| > 0.5, p = 2$)

Fig. 4. Results of four methods according to the number of selected genes.

5. Concluding remarks

Microarray data make up of many gene expression data and relatively small samples. As we include more genes, misclassification can increase. It means that all gene expression data are not related to the difference between the two cancer types: ALL and AML. Thus it is necessary to identify predictive genes. The proposed genetic filter identified a subset of predictive genes and bettered prediction quality over previous prediction models. Although the proposed genetic filter performed well, there is room for further improvement. Clustering techniques, e.g., K-means, DBScan, or Herd Clustering [29], may remove the redundancy of gene expression data before the proposed genetic filter is applied. We will leave this for future work.

Acknowledgments

A preliminary version of this paper appeared in the Proceedings of the Genetic and Evolutionary Computation Conference, Seattle, Washington, USA, 2004, pp. 346–355. The present research has been conducted by the Research Grant of Kwangwoon University in 2014. This research was supported by the Gachon University research fund of 2015(GCU-2015-0030). The authors thank Dr. Su-Yeon Lee and Prof. Byung-Ro Moon for their valuable suggestions in improving this paper.

References

[1] P. O. Brown and D. Botstein, Exploring the new world of the genome with DNA microarrays, Nature Genetics **21** (1999), 33–37.

[2] D. J. Lockhart and E. A. Winzeler, Genomics, gene expression and DNA arrays, Nature **405** (2000), 827–836.

[3] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield and E. S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science **286** (1999), 531–537.

[4] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack and A. J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proceedings of the National Academy of Sciences of the United States of America **96** (1999), 6745–6750.

[5] A. Ben-Dor, R. Shamir and Z. Yakhini, Clustering gene expression patterns, Journal of Computational Biology **6** (1999), 281–297.

[6] M. Bittner, P. Meltzer and J. Trent, Data analysis and integration: of steps and arrows, Nature Genetics **22** (1999), 213–215.

[7] G. Getz, E. Levine, E. Domany and M. Q. Zhang, Superparamagnatic clustering of yeast gene expression profiles, Physica A **279** (2000), 457–464.

[8] E. Hartuv, A. O. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrach and R. Shamir, An algorithm for clustering cDNA fingerprints, Genomics **66** (2000), 249–256.

[9] G. Getz, E. Levine and E. Domany, Coupled two-way clustering analysis of gene microarray data, Proceedings of the National Academy of Sciences of the United States of America **97** (2000), 12079–12084.

[10] F. Liu and L. Wang, Biclustering of time-lagged gene expression data using real data, Journal of Biomedical Science and Engineering **3** (2010), 217–220.

[11] F. Liu, Time-lagged co-expression gene analysis based on biclustering technology, Biotechnology & Biotechnological Equipment **27** (2013), 4031–4039.

[12] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer and Z. Yakhini, Tissue classification with gene expression profiles, Journal of Computational Biology **7** (2000), 559–583.

[13] L. Li, T. A. Darden, C. R. Weinberg and L. G. Pedersen, Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbor method, Combinatorial Chemistry & High Throughput Screening **4** (2001), 727–739.

[14] I. Guyon, J. Weston, S. Barnhill and V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning **46** (2002), 389–422.

[15] S. Ando and H. Iba, Artificial immune system for classification of gene expression data, In Proceedings of the Genetic and Evolutionary Computation Conference, Chicago, Illinois, USA, 2003, pp. 1926–1937.

[16] Y.-H. Kim, S.-Y. Lee and B.-R. Moon, A genetic approach for gene selection on microarray expression data, In Proceedings of the Genetic and Evolutionary Computation Conference, Seattle, Washington, USA, 2004, pp. 346–355.

[17] A. H. Chena and C.-H. Lin, A novel support vector sampling technique to improve classification accuracy and to identify key genes of leukaemia and prostate cancers, Expert Systems with Applications **38** (2011), 3209–3219.

[18] S. Das and A. K. Das, An approach towards most cancerous gene selection from microarray data, Computational Intelligence in Data Mining - Volume 3, Smart Innovation, Systems and Technologies **33** (2015), 641–648.

[19] S. Penga, Q. Xub, X. B. Lingc, X. Pengd, W. Dua and L. Chen, Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines, FEBS Letters **555** (2003), 358–362.

[20] S. Sun, Q. Peng and A. Shakoor, A kernel-based multivariate feature selection method for microarray data classification, PLoS ONE **9** (2014), e102541.

[21] F. Yang and K. Z. Mao, Robust feature selection for microarray data based on multicriterion fusion, IEEE/ACM Transactions on Computational Biology and Bioinformatics **8** (2011), 1080–1092.

[22] X. Zhang and H. Ke, ALL/AML cancer classification by gene expression data using SVM and CSVM approach, Genome Informatics **11** (2000), 237–239.

[23] J. Yang and V. Honavar, Feature subset selection using a genetic algorithm, Feature Extraction, Construction and Selection – The Springer International Series in Engineering and Computer Science **453** (1998), 117–136.

[24] K.-C. Wong, K.-S. Leung and M.-H. Wong, An evolutionary algorithm with species-specific explosion for multimodal optimization, In Proceedings of the Genetic and Evolutionary Computation Conference, Montreal, Canada, 2009, pp. 923–930.

[25] K.-C. Wong, C.-H. Wu, R. K.P. Mok, C. Peng and Z. Zhang, Evolutionary multimodal optimization using the principle of locality, Information Sciences **194** (2012), 138–170.

[26] Y.-H. Kim and B.-R. Moon, Lock-gain based graph partitioning, Journal of Heuristics **10** (2004), 37–57.

[27] B. Efron, The Jacknife, the Bootstrap, and Other Resampling Plans, Society for Industrial and Applied Methematics, 1982.

[28] B. Efron and R. Tibshirani, Cross-validation and the bootstrap: estimating the error rate of a prediction rule, Technical Report 176, Dept. of Statistics, Stanford University, 1995.

[29] K.-C. Wong, C. Peng, Y. Li and T.-M. Chan, Herd Clustering: a synergistic data clustering approach using collective intelligence, Applied Soft Computing **23** (2014), 61–75.