Available online at www.sciencedirect.com

ScienceDirect

Information Sciences 178 (2008) 88–105

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

Direct integration of microarrays for selecting informative genes and phenotype classification

Youngmi Yoon [a,c], Jongchan Lee [a], Sanghyun Park [a,*], Sangjay Bien [a], Hyun Cheol Chung [b], Sun Young Rha [b]

[a] Department of Computer Science, Yonsei University, 134 Sinchon-dong, Seodaemun-gu, Seoul 120-749, South Korea
[b] Department of Internal Medicine, Cancer Metastasis Research Center, Yonsei University College of Medicine, South Korea
[c] Department of Information Technology, Gachon University of Medicine and Science, South Korea

Received 7 February 2007; received in revised form 25 July 2007; accepted 1 August 2007

Abstract

The ability to provide thousands of gene expression values simultaneously makes microarray data very useful for phenotype classification. A major constraint in phenotype classification is that the number of genes greatly exceeds the number of samples. We overcame this constraint in two ways; we increased the number of samples by integrating independently generated microarrays that had been designed with the same biological objectives, and reduced the number of genes involved in the classification by selecting a small set of informative genes. We were able to maximally use the abundant microarray data that is being stockpiled by thousands of different research groups while improving classification accuracy. Our goal is to implement a feature (gene) selection method that can be applicable to integrated microarrays as well as to build a highly accurate classifier that permits straightforward biological interpretation. In this paper, we propose a two-stage approach. Firstly, we performed a direct integration of individual microarrays by transforming an expression value into a rank value within a sample and identified informative genes by calculating the number of swaps to reach a perfectly split sequence. Secondly, we built a classifier which is a parameter-free ensemble method using only the pre-selected informative genes. By using our classifier that was derived from large, integrated microarray sample datasets, we achieved high accuracy, sensitivity, and specificity in the classification of an independent test dataset.

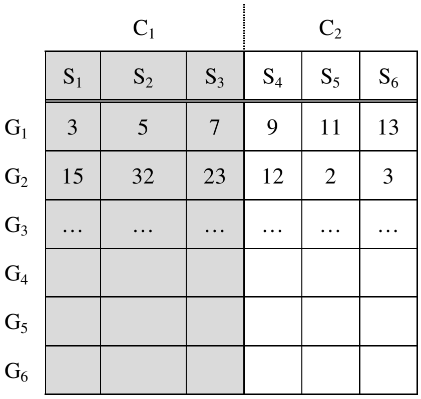
Keywords: Data mining; Microarray data analysis; Microarray data integration; Microarray data classification; Informative gene selection

1. Introduction

Recently researchers have examined tumor cell specific gene expression patterns and have made use of the molecular characteristics of tumor tissue for diagnostic purposes. Since microarray technology is capable of

[*] Corresponding author. Tel.: +82 2 2123 5714; fax: +82 2 365 2579.
E-mail address: sanghyun@cs.yonsei.ac.kr (S. Park).

0020-0255/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2007.08.013



	C$_1$			C$_2$		
	S$_1$	S$_2$	S$_3$	S$_4$	S$_5$	S$_6$
G$_1$	3	5	7	9	11	13
G$_2$	15	32	23	12	2	3
G$_3$	…	…	…	…	…	…
G$_4$						
G$_5$						
G$_6$						

Fig. 1. Organization of microarray data. (S_i is a sample, G_i is a gene, while C_1 and C_2 are each class labels.)

screening thousands of genes simultaneously, microarray data is expected to bring about drastic advances in tumor diagnosis. As shown in Fig. 1, microarray data are organized as matrices with each column representing a sample, each row representing a gene, and each cell representing the particular gene expression value within a particular sample. Since the simultaneous measurement of expression levels using tens of thousands of probes is now feasible, statistical methods are required for the analysis and interpretation of this large volume of data.

When using statistical methods, increasing sample size is usually desirable for obtaining more reliable classification results. Performing an analysis with a large sample is essential for deducing a meaningful conclusion from the data when working on tumor-related research. Recently, Rhodes [17] performed a meta-analysis of multiple datasets that addressed a similar hypothesis. His meta-analysis was used to validate and statistically assess all of the positive results simultaneously.

Even when considering only microarray data with the same experimental objectives, difficulties in integrating microarray data across experiments can arise from microarray platform differences, gene sets, and technology and protocol variation between labs. Deciding how to combine the data on the gene expression level in different microarrays is a challenging problem because gene expression levels measured from different experiments are not necessarily directly comparable. In this paper, we propose a method to integrate independent microarray datasets and to build a classifier through two stages.

Firstly, we combined the integration algorithms and the filtering methods and used them to select a set of informative genes. Our integration algorithms do not require massive computation for normalization. Our informative gene filtering algorithm is a rank-based approach within each sample. In the second stage of the process, we built a classifier using only the pre-selected informative genes. The biological interpretation of our classifier is relatively simple. Our classifier consists of K ($\geqslant$5) rules where each rule has a relationship among three genes and a class label. Since this second stage of classifier building uses only the pre-selected genes relevant to the classification, our classifier is capable of increasing classification accuracy while offering affordable computation times even for integrated microarray datasets of large sample size. Experimental results showed that our system was able to classify with better accuracy than conventional approaches as the sample size of the training datasets grew larger. Our two-stage system effectively maximizes the use of the accumulated independent microarray datasets and sheds light on a new paradigm in the field of microarray data integration.

2. Related work

2.1. Microarray data integration

Several methods have been proposed previously for integrating microarray data. One method uses a meta-mining technique [4] in which individually obtained microarray experiment results are integrated and analyzed. However, because the sample size within each individual experiment is generally small, the experimental results themselves are not reliable and the integration of these results may produce an even worse analysis. Another method of integration is to normalize data obtained from individual research into values derived from a common scale and then combine them [10]. The most representative example of this method involves transforming the data to Z-Scores before combining them. However, this method, which also involves a massive normalization process, involves considerable computation during the preprocessing stage. Studies presenting data integration models other than those mentioned above also include a method [13] that uses a correlation signature for integrating heterogeneous microarray data.

2.2. Informative gene selection

The greatest restriction in analyzing microarray data is that the number of genes is far bigger than the number of samples involved in the experiment [6]. In reality, however, the number of genes that affect classification is very limited. Thus, most genes are "noise" genes that do not affect class discrimination. Informative genes, as shown in Fig. 2, can be defined as genes with high expression values on the whole within Class C_1 and low expression values on the whole within Class C_2. On the other hand, genes that do not provide a consistent level of expression values for specific classes can be regarded as noise genes that do not have any relevance [23,24]. Therefore, a rational method is to firstly identify only the relevant genes that participate in phenotype identification for specific diseases, then come up with the classification method using only those genes.

The process of eliminating genes not associated with the disease phenotype while also identifying only the informative genes is called the *feature selection*, and it is very important for microarray data analysis [2]. Currently, various methods can be used to precisely and effectively select these informative genes. Linear combination method like the PCA (Principal Component Analysis) [3] is typical of the methods used in feature selection. The PCA method does reduce the dimension of microarray data by using eigen vectors, but this method does not individually find genes that are relevant to classification. Another typical feature selection method, the parametric method, assumes a statistical model representing the data, like the *t*-statistic or the Golub [8] method. This method saves parameters (e.g. mean and variance) that can represent the model. Since the parametric method replaces thousands of gene expression values with a very small number of parameters, information loss could possibly become a problem. On the other hand, the non-parametric method [2,18] aligns all sample values of a single gene and calculates the score (degree of interruption for a complete separation) representing how much that gene was differently expressed between the two class groups. When the gene is considered as a feature, the most commonly used feature selection method is the score-based approach. The score-based feature selection method measures in statistical values how much more significant each feature is compared to the other features, then sorts them, and selects the top ranked features. The most popular score-based feature selection methods are the Information Gain [9,27], Relief-F [20], and the method using correlation coefficient [1].

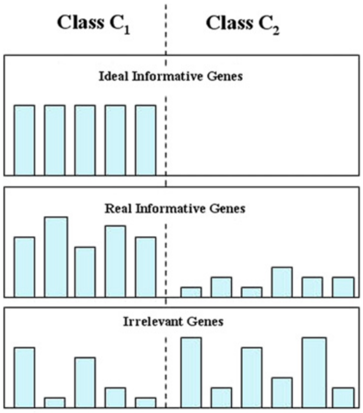


Fig. 2. An informative gene viewed as a type of expression value.

The Information Gain method is a popular feature filtering algorithms to rank features. This method has been used for gene selection in microarray data by Li [16]. It measures the number of bits of information obtained for class prediction by calculating the value for each feature (gene). The gene expression value is continuous-valued, and Information Gain requires that numeric features be discretized. We used the entropy-based discretization method [7,9] implemented in Weka [27]. Let D be a data set of class-labeled samples. There are two (Tumor, Normal) distinct classes, C_i (for $i = 1,2$). Let $C_{i,D}$ be the set of samples of class C_i in D. Let $|D|$ and $|C_{i,D}|$ denote the number of samples in D and $C_{i,D}$, respectively

$$Info(D) = -\sum_{i=1}^{2} p_i \log p_i,$$

where p_i is estimated by $|C_{i,D}|/|D|$. Gene A can be used to split D into v partitions, $\{D_1, D_2, \ldots, D_v\}$, where D_j contains those samples in D that have outcome a_j of A

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} * Info(D_j).$$

Then, the Information Gain value for gene A is

$$Gain(A) = Info(D) - Info_A(D).$$

After computing these values for all of the genes, the genes that have high information gain values are selected as informative genes.

The basic idea behind the Relief-F method is that each gene's weight is calculated by finding the F closest samples (half from the same class (hit) and others from another class (miss)) to each sample. If A is a selected sample and G is a selected gene, the weight of the gene is increased, in the case of a hit, by the distance between A and the hit sample. In contrast, the weight is decreased, in the case of a miss, by the distance between A and the missed sample. After performing these computations and aggregations for all genes, the k-genes with the highest weight are selected.

Park's method [18] builds a binary sequence for a gene and uses Kendall's correlation coefficient to calculate a score that measures how differently the genes are expressed in the two class groups [1].

All of the methods mentioned above use the expression value of each gene as it is, without any consideration regarding the integration of the microarray data.

2.3. Classification

The SVM [11,12,25] and the k-Nearest Neighbor [5] methods are more commonly used among numerous classification approaches. The SVM is based on a machine learning algorithm that functions by learning linear decision rules, which are represented by hyper planes. The SVM is not only used in microarray classification, but is also used in other areas, such as regression analysis and density prediction. The SVM is complicated to apply to microarray data because it experimentally needs various types of parameter adjustments. The k-Nearest Neighbor (k-NN) [5] is an algorithm that classifies samples by selecting similar ones from the individual training dataset of new samples. This k-NN algorithm has the weakness of not providing a good efficiency when granting equal weights to all genes.

Some classification methods do not use parameters, but instead adopt a data-driven machine learning approach that was proposed by Tan and called the k-TSP (Top Scoring Pair) [22] method. The TSP is an algorithm that finds a pair of genes with the highest score. For each gene pair, X_i, X_j, the score is the difference of the relative frequencies of occurrences when $X_i < X_j$ in each class. The k-TSP classifier consists of k top-scoring gene pairs that achieve the high score. Two drawbacks of the pair-gene rule are the possibility that the two genes could be selected by chance alone and that a small alteration in the training datasets might change the top scoring gene pair. The k-TSP method builds a classifier without the step of extracting the informative genes. Since all of the genes in the microarray datasets are employed in the classification stage, this method is computationally expensive as the microarray datasets are getting integrated.

3. System overview

The system overview is shown in Fig. 3. In the first stage, the integration of independently generated microarray datasets is accomplished. Each independent microarray dataset has different probe sets and a distinct variation in the scale of expression values. Only the genes common to all microarray datasets are extracted. The expression value of each sample in each experiment is transformed into a rank value within a sample. Once the expression values are changed to rank values, the integration of samples originating from different experiments becomes feasible, as long as their gene order is the same. After integration, a score that measures how differently a gene is expressed in the two class groups is calculated for each gene. At this stage, genes with a very small score or a very large score are candidates for informative genes. In the second stage, a classifier is built by using only the informative genes that were selected in stage 1.

As a first attempt to generalize the number of genes involved in a rule and to solve the drawbacks of the pair-gene rule, we propose the k-GeneTriple method. For each set of three genes, X_i, X_j, X_k, one can establish six (3!) magnitude relationships by comparing the rank values of the three genes. For each relationship, among the samples with a class C_1 label, the number of samples that satisfy the relationship is divided by the number of C_1 samples and then saved. Likewise, among the samples with a class C_2 label, the number of samples that satisfy the relationship is divided by the number of C_2 samples and then saved. For every relationship, the difference between these two values is calculated. A relationship with a higher difference score represents a more discriminative classification rule. The classifier consists of top k classification rules. This parameter, k, is determined by applying LOOCV (Leave One Out Cross Validation) to the training dataset. The LOOCV method is an approach that uses all the samples except one sample in a microarray dataset, builds a classifier, and measures the classifier's accuracy by applying it to the single sample that was excluded. Each classification rule consists of a set of three genes, the magnitude relationship among those three genes, and the prevalent class label for the relationship. Given a new test sample, one can apply the classifier to the sample, predict the class label of the test sample by a majority vote, and compare this predicted class with the real class of the test sample.

4. System implementation

This section describes the details of the two-stage processing algorithm on the microarray data. In Section 4.1, the integration procedure of microarray datasets and informative gene selection algorithm are presented.

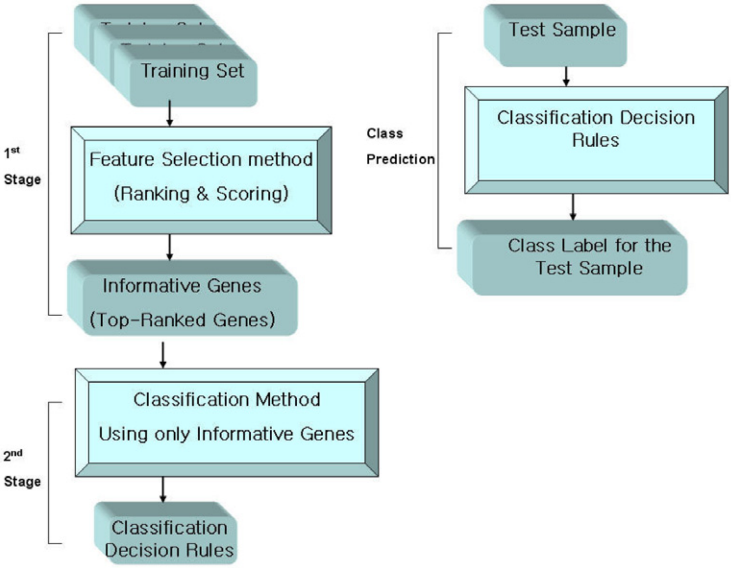


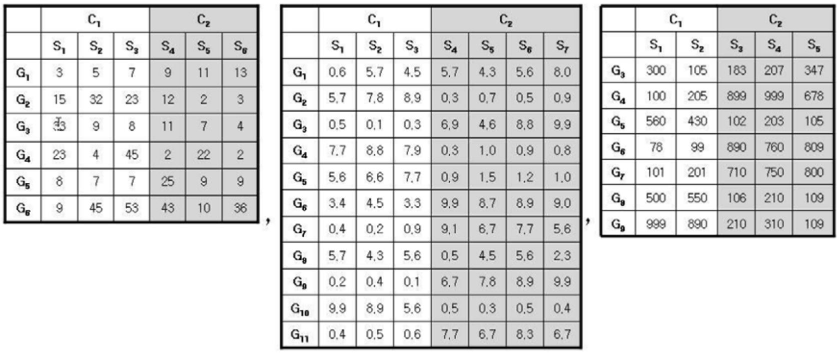
Fig. 3. Overview of our system.

In Section 4.2, we present the *k*-GeneTriple classification method, which compares the magnitude relationship among three genes, converts the relationship into a score, and builds a classifier which consists of *k* gene triples.

4.1. Microarray data integration and informative gene selection

From among the microarray datasets that were generated independently but which had the same experimental objectives (Fig. 4), only the genes common to all datasets were extracted (Fig. 5).

Even if the set of common genes has the same order in all of the microarray datasets, the scale of the expression values for each set of microarray data may be quite different because of different experimental conditions or protocols. In these cases, a direct integration is inappropriate. Therefore, in order to make the direct integration possible, we used the rank of expression value for the corresponding gene within each sample rather than using the actual expression values. Accordingly, as shown in Fig. 6, the expression values were all converted into ranks within each sample. Our system uses the rank of expression value for the corresponding gene within each sample, sorts the rank levels from the smallest to the largest for each gene along with the class label of each sample (0 for normal, 1 for tumor), and calculates the score, which is the number of swaps between neighboring 0 and 1 values. Table 1 shows an algorithm for identifying informative genes. To help understand the algorithm, let us assume the microarray data is as presented in Table 2 below. By changing this data into rank-based within each sample, we produce the values shown in Table 3. Then by sorting the rank levels from the smallest to the largest for each gene along with the class label of the sample, we create



	C_1			C_2		
	S_1	S_2	S_3	S_4	S_5	S_6
G_1	3	5	7	9	11	13
G_2	15	32	23	12	2	3
G_3	33	9	8	11	7	4
G_4	23	4	45	2	22	2
G_5	8	7	7	25	9	9
G_6	9	45	53	43	10	36

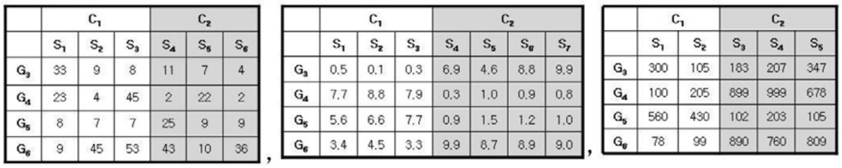
,

	C_1			C_2			
	S_1	S_2	S_3	S_4	S_5	S_6	S_7
G_1	0,6	5,7	4,5	5,7	4,3	5,6	8,0
G_2	5,7	7,8	8,9	0,3	0,7	0,5	0,9
G_3	0,5	0,1	0,3	6,9	4,6	8,8	9,9
G_4	7,7	8,8	7,9	0,3	1,0	0,9	0,8
G_5	5,6	6,6	7,7	0,9	1,5	1,2	1,0
G_6	3,4	4,5	3,3	9,9	8,7	8,9	9,0
G_7	0,4	0,2	0,9	9,1	6,7	7,7	5,6
G_8	5,7	4,3	5,6	0,5	4,5	5,6	2,3
G_9	0,2	0,4	0,1	6,7	7,8	8,9	9,9
G_{10}	9,9	8,9	5,6	0,5	0,3	0,5	0,4
G_{11}	0,4	0,5	0,6	7,7	6,7	8,3	6,7

,

	C_1		C_2		
	S_1	S_2	S_3	S_4	S_5
G_3	300	105	183	207	347
G_4	100	205	899	999	678
G_5	560	430	102	203	105
G_6	78	99	890	760	809
G_7	101	201	710	750	800
G_8	500	550	106	210	109
G_9	999	890	210	310	109

Fig. 4. Three of the independently generated microarray dataset for prostate cancer.



	C_1			C_2		
	S_1	S_2	S_3	S_4	S_5	S_6
G_3	33	9	8	11	7	4
G_4	23	4	45	2	22	2
G_5	8	7	7	25	9	9
G_6	9	45	53	43	10	36

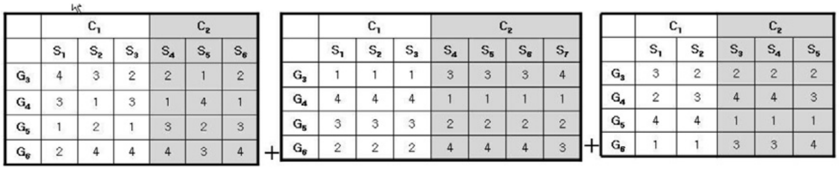
,

	C_1			C_2			
	S_1	S_2	S_3	S_4	S_5	S_6	S_7
G_3	0.5	0.1	0.3	6.9	4.6	8.8	9.9
G_4	7.7	8.8	7.9	0.3	1.0	0.9	0.8
G_5	5.6	6.6	7.7	0.9	1.5	1.2	1.0
G_6	3.4	4.5	3.3	9.9	8.7	8.9	9.0

,

	C_1		C_2		
	S_1	S_2	S_3	S_4	S_5
G_3	300	105	183	207	347
G_4	100	205	899	999	678
G_5	560	430	102	203	105
G_6	78	99	890	760	809

Fig. 5. Extraction of the set of common genes among microarray datasets.



	C_1			C_2		
	S_1	S_2	S_3	S_4	S_5	S_6
G_3	4	3	2	2	1	2
G_4	3	1	3	1	4	1
G_5	1	2	1	3	2	3
G_6	2	4	4	4	3	4

+

	C_1			C_2			
	S_1	S_2	S_3	S_4	S_5	S_6	S_7
G_3	1	1	1	3	3	3	4
G_4	4	4	4	1	1	1	1
G_5	3	3	3	2	2	2	2
G_6	2	2	2	4	4	4	3

+

	C_1		C_2		
	S_1	S_2	S_3	S_4	S_5
G_3	3	2	2	2	2
G_4	2	3	4	4	3
G_5	4	4	1	1	1
G_6	1	1	3	3	4

Fig. 6. Microarray data expressed as ranks within each sample.

Table 1
Informative gene selection algorithm

Input	NI (the number of informative genes), $V[][]$ (expression values)
Output	IG[][] (informative genes)
1	Generate a binary sequence S, which replaces normal samples with 0 and tumor samples with a value of 1
2	For all i, j, replace $V[G_i][S_j]$, which represents an expression value, with $R[G_i][S_j]$, which represents the order when they are ranked according to expression values within each sample
3	Select an arbitrary gene G_i among genes that were not selected
4	For all j, sort $R[G_i][S_j]$ in ascending order and generate a binary sequence, T, where normal samples are replaced with 0 and tumor samples are replaced with 1
5	Using the scoring function defined as the number of swaps, calculate the scores for S and T, and insert the score for T into a priority queue with size NI
6	Repeat step 3 until there are no unselected genes left
7	From the priority queue, select half of NI number of informative genes from the top (front), and half of NI number of informative genes from the bottom (rear)

Table 2
Data presented in expression value

	Normal	Normal	Normal	Tumor	Tumor	Tumor
G_1	13	32	3	24	13	42
G_2	25	12	26	3	1	2
G_3	23	6	2	102	59	13
G_4	7	20	63	4	7	27

Table 3
Data presented in rank

	Normal	Normal	Normal	Tumor	Tumor	Tumor
G_1	2	4	2	3	3	4
G_2	4	2	3	1	1	1
G_3	3	1	1	4	4	2
G_4	1	3	4	2	2	3

the data presented in Table 4. Finally, if we change the class label of a sample into binary sequence, the data shown in Table 5 is produced.

After carrying out step 1 as outlined in Table 1, the initial binary sequence S becomes “000111”, which is a perfect splitting. When we run the function which calculates the score as the number of swaps of consecutive

Table 4
Data after sorting

	N or T	N or T	N or T	N or T	N or T	N or T
G_1	2 (N)	2 (N)	3 (T)	3 (T)	4 (N)	4 (T)
G_2	1 (T)	1 (T)	1 (T)	2 (N)	3 (N)	4 (N)
G_3	1 (N)	1 (N)	2 (T)	3 (N)	4 (T)	4 (T)
G_4	1 (N)	2 (T)	2 (T)	3 (N)	3 (T)	4 (T)

Table 5
Data expressed as binary sequence

G_1	0	0	1	1	0	1
G_2	1	1	1	0	0	0
G_3	0	0	1	0	1	1
G_4	0	1	1	0	1	1

0s and 1s to arrive at S (perfect splitting) for each gene in Table 5, the gene with the smallest score is G3, with a total of 1 time, and the gene with the largest score is G2 with a total of 9 times. This means that G2 and G3 have a strong possibility of becoming informative genes than G1 or G4.

4.2. *k*-GeneTriple classification method

In this paper we are attempting to generalize the number of genes involved in each rule, such that a classifier has k-rules, where some of the rules involve two genes while others involve three or more genes, in order to increase the robustness and the reliability of the classifier for tumor and normal sample class prediction. As the first step, we propose the k-GeneTriple method.

In the k-GeneTriple method, the number of genes involved in a classification rule is limited to three. For each set of three genes, we establish six magnitude relationships like R_1, R_2, R_3, R_4, R_5, R_6 in Table 7. For each relationship, we calculate the score, which is the difference between the probability that the relationship occurs in class 1 and the probability that the relationship occurs in class 2. The set of three genes satisfying the relationship with higher score is regarded as the more discriminative for classification. Each relationship also keeps its class label by comparing the two probabilities and adopting the class that has more prevalent probability. We calculate the scores for all three gene sets and for all six magnitude relationships for each set. These scores are placed into a priority queue in descending order. We take the k relationships that have the higher score. Our classifier consists of k classification rules and each classification rule consists of (1) a set of three genes, (2) the magnitude relationship among those three genes, and (3) the class label of the relationship. Table 6 shows an algorithm for the k-GeneTriple method.

In addition, the scoring function used in step 3 of Table 6 is as follows:

$P_{ijk}(1)$	The probability that a relationship of $X_i < X_j < X_k$ occurs in class 1 (X_i, X_j, X_k stand for the rank values within a sample)
Δ_{ijk}	$\lvert P_{ijk}(1) - P_{ijk}(2)\rvert$

For example, let us assume the dataset shown in Table 7. Here, when all of the corresponding values of Δ are calculated, in the case of the R_3 ($X_j < X_i < X_k$) relationship, one can see that

$$\Delta_{jik} = |P_{jik}(1) - P_{jik}(2)| = |29/42 - 1/33| \approx 0.66$$

Table 6
k-GeneTriple classification algorithm

Input	K (the number of rules specified), IS[][] (informative genes)
Output	A set of K number of classification rules
1	From the informative gene set, select a set of three genes that were not processed before
2	Enumerate the magnitude relationships among the three genes for all samples
3	Calculate the score for each three gene combination using the scoring function
4	Insert the rule which is composed of the calculated score, the gene combination, the magnitude relationship, and the class label of this gene combination into the priority queue with size K
5	Repeat step 1 if there are three gene combinations that have not been processed
6	Select the top K rules from the priority queue

Table 7
k-GeneTriple example

	R_1	R_2	R_3	R_4	R_5	R_6	Total
C_1	2	1	29	4	2	4	42
C_2	4	5	1	14	8	1	33

(Definition of R) R_1: $X_i < X_j < X_k$, R_2: $X_i < X_k < X_j$, R_3: $X_j < X_i < X_k$, R_4: $X_j < X_k < X_i$, R_5: $X_k < X_i < X_j$, R_6: $X_k < X_j < X_i$.

has the largest score. This means that if one observes the R_3 relationship in a given test sample, then the sample class is predicted to be C_1. Actually, we applied k number of decision rules to the test sample, and performed majority voting to predict the class of the sample. The value of k is determined by LOOCV.

Majority voting is an ensemble approach [19] which combines the prediction (predicted class) of multiple rules, and obtain the final prediction; in other words the classifier simply chooses the class receiving the most votes. Our classifier consists of k decision rules. Let the each rule be r_i. $L(r_i)$ is the class label of the rule r_i. $P_{ri}(S)$ is the predicted class when the sample S satisfies the decision rule r_i. The value which $P_{ri}(s)$ can have is either "Normal" or "Tumor". For ease of computation, the value of $P_{ri}(S)$ is converted into $V(r_i)$ whose value is 1 when the value of $P_{ri}(S)$ is "Normal" and 0 when the value of $P_{ri}(S)$ is "Tumor". NC is the unweighted sum of the $V(r_i)$ for all i. If the value of NC is larger than the half of the number of rules ($k/2$), then S is finally predicted to be a normal sample, otherwise the sample is predicted to be a tumor sample. Since we fixed the number of rules to be an odd number, our system can break the tie and always return a predicted class label. The majority voting process is described in short as follows:

- r_i is the ith rule
- k is the number of rules
- S is a test sample
- NC is the number of normal cell count

$$L(r_i) = \text{Class Label of the } r_i \ (L(r_i) \in \{\text{normal}, \text{tumor}\})$$

$$P_{ri}(S) = \begin{cases} L(r_i) & \text{if } S \text{ satisfies the } r_i \\ \overline{L(r_i)} & \text{Otherwise} \end{cases}$$

$$V(r_i) = \begin{cases} 1 & \text{if } P_{ri}(S) \text{ is a Normal Sample} \\ 0 & \text{Otherwise} \end{cases}$$

$$\text{NC} = \sum_{i=1}^{k} V(r_i)$$

5. Experimental results

In this section, we describe the experiments we performed to verify the accuracy and efficiency of the two-stage method. We used publicly available prostate cancer microarray data. The platform of these data was the Affymetrix HG_95AV2. For convenience, we represent each dataset as an abbreviation of the first author of the published papers by Singh [21], Welsh [26] and LaTulippe [14]. Table 8 shows the information about the microarray datasets that were used in our experiment. Section 5.3.2 describes the accuracy of the classification using colon cancer microarray data of which platform is cDNA.

5.1. Determining the optimal number of rules (k) by LOOCV

In this subsection, we describe the experiment that determined the optimal number of rules which is the value of k, by LOOCV. We varied the value of k and chose the k value that gave the highest LOOCV accuracy in each dataset. Since most of the previously proposed gene ranking methods typically select 50–200 top-ranked genes [8,15], we fixed the number of informative genes as 126, 1% of 12600 genes that is the number of common genes in the microarray data. In order to break ties in the majority voting procedure, we imposed a restriction that k does not exceed 10 and is an odd number in LOOCV experiments. Table 9 shows the summary of the optimal k value obtained from the experiments in each training dataset. We measured accuracy,

Table 8
Prostate microarray data

Data	Number of probes	Number of normal samples	Number of tumor samples	Total number of samples
Singh	12600	50	52	102
Welsh	12626	9	24	33
LaTulippe	12626	3	23	26

Table 9
The optimal k values

Training dataset	Optimal k
Singh	9
LaTulippe	5
Welsh	5
Singh + Welsh	9
Singh + LaTulippe	7
Welsh + LaTulippe	5

sensitivity, and specificity in order to compare our system's performance with others' performance. These measures were defined as follows:

$$\text{Accuracy} = \frac{\text{The Number of Correctly Predicted Samples}}{\text{The Number of Total Samples}},$$

$$\text{Sensitivity} = \frac{\text{The Number of Correctly Predicted Tumor Samples}}{\text{The Number of Tumor Samples}},$$

$$\text{Specificity} = \frac{\text{The Number of Correctly Predicted Normal Samples}}{\text{The Number of Normal Samples}}.$$

In this experiment, the number of rules was restricted to no less than 5. If the number of rules is too small, the rules cannot guarantee credibility as a classifier. Moreover, independent test data may not contain the genes involved in the classifier.

5.2. Accuracy of the informative gene selection method

In this subsection, we describe the accuracy test of the proposed informative gene identification method. We compared our gene selection method with the Information Gain and Relief-F methods, which are popular feature filtering methods. Since these two methods cannot be directly applied to integrated data, we normalized all of the data by applying a Z-score, which is a classic, but the most generalized normalization method. After selecting the informative genes by individually using our proposed method, the Information Gain method, and the Relief-F method, we compared the accuracy of those three gene identification methods by applying a classification method. For the classification method, we used the linear support vector machine (SVM). The size of the informative gene is 1% of the original genes.

Figs. 7–9 present the accuracy of each of the independent Singh, Welsh, and LaTulippe datasets individually. We used one dataset as independent test data and used the other two datasets as training data. When using Singh as the independent test data, the training datasets were Welsh, LaTulippe, and Welsh + LaTulippe. We built a classifier from each training dataset and applied the classifier to Singh and measured the accuracy. The experimental accuracy results for Singh were compared among different training datasets. We also compared the accuracy of the experimental results for Singh among three different gene selection methods: (1) our proposed gene selection method + SVM, (2) Relief-F + SVM, and (3) Information Gain + SVM.

The classification accuracy on independent test data confirmed that the proposed informative gene identification method has a comparable or better performance than the Information Gain and Relief-F methods when integrated dataset of Singh and LaTulippe were used as the test dataset. In addition, the classification accuracy of our method increased as the sample size in the training datasets is increased by data integration.

5.3. Accuracy of the classification method

In this section, we tested the accuracy of our classification method (k-GeneTriple) using the optimal k that was acquired from Section 5.1. We compared our system with other two methods: (1) k-TSP, (2) Information Gain + SVM. The Relief-F method was excluded in this experiment because it showed a worse accuracy than

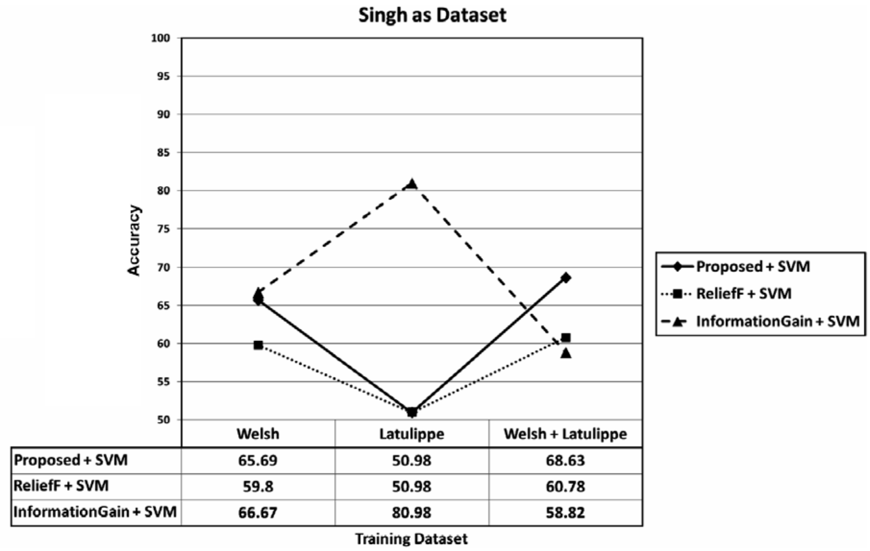


Fig. 7. Accuracy when Singh was used as test data.

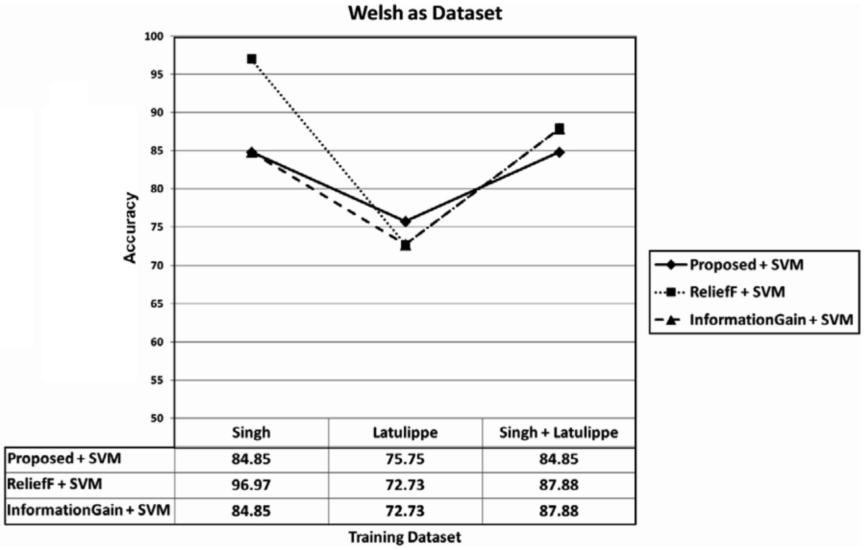


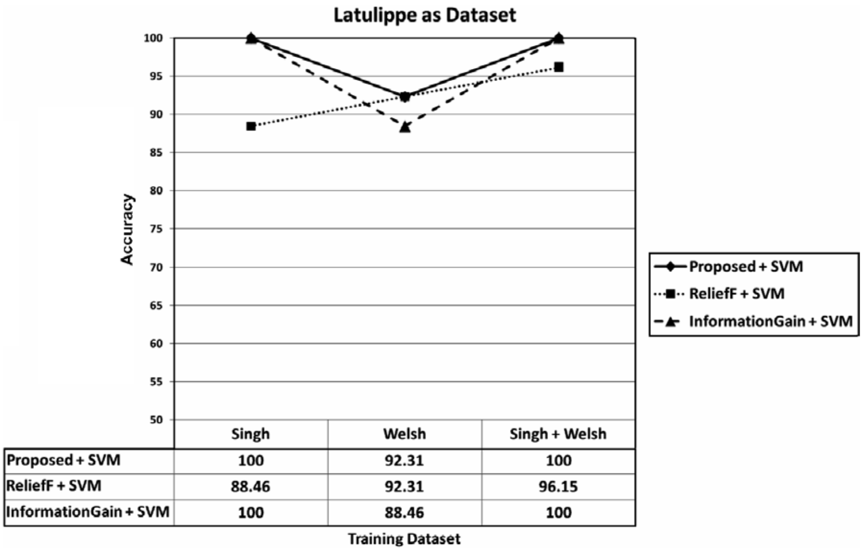
Fig. 8. Accuracy when Welsh was used as test data.

the Information Gain method overall as proved in Section 5.2. Section 5.3.1 presents the accuracy analysis using Affymetrix prostate cancer microarray data, and Section 5.3.2 presents the accuracy analysis using cDNA colon cancer microarray data.

5.3.1. Accuracy of the classification method using Affymetrix data

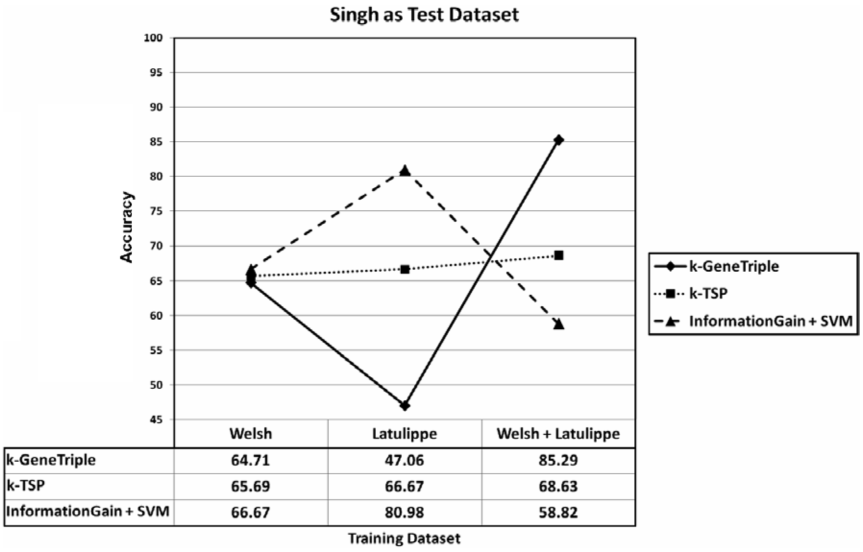
As shown in Figs. 10–12, we built a classifier using a training dataset, which consists of all possible data combinations, excluding the test dataset, and measured the accuracy of each independent test dataset. Table 10 shows the values of optimal k used in our experiments.

As seen in the above figures, the experimental results of our system did not always perform better than other systems when the training dataset is a single microarray dataset. However, this decrease in performance is partly due to the small sample size in the single microarray dataset. In particular, both the Welsh and



	Singh	Welsh	Singh + Welsh
Proposed + SVM	100	92.31	100
ReliefF + SVM	88.46	92.31	96.15
InformationGain + SVM	100	88.46	100

Fig. 9. Accuracy when LaTulippe was used as test data.



	Welsh	Latulippe	Welsh + Latulippe
k-GeneTriple	64.71	47.06	85.29
k-TSP	65.69	66.67	68.63
InformationGain + SVM	66.67	80.98	58.82

Fig. 10. Accuracy when Singh was used as test data.

LaTulippe datasets had skewed samples with a much smaller number of normal samples than tumor samples. Therefore, these datasets cannot be used as training datasets alone. However, as integration significantly increases the sample size, our system performed with a much better accuracy than the k-TSP and SVM methods. Based on these experiments, the proposed two-stage approach can produce more credible classifiers than other systems, especially when data are comprehensively integrated.

5.3.2. *Accuracy of the classification method using cDNA microarray*

The microarray data we used for this experiment was a colon cancer study from the Cancer Metastasis Research Center of Yonsei University [28]. The platform of these data was cDNA. Details about these data are given in Table 11. The two microarray batches were made with time delays, an A batch and a B batch. The gene sets in these two batches were the same. ‘‘Paired’’ means that one normal tissue sample and one tumor

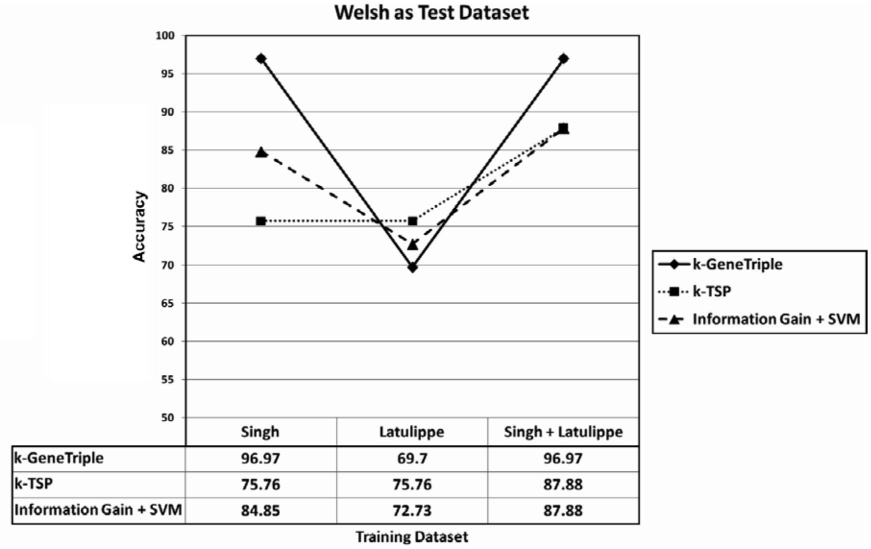


Fig. 11. Accuracy when Welsh was used as test data.

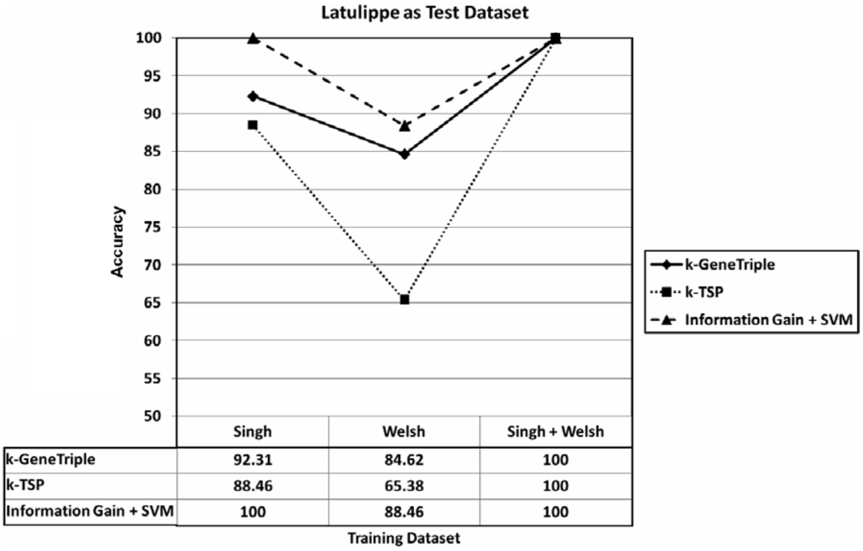


Fig. 12. Accuracy when LaTulippe was used as test data.

Table 10
The values of optimal k used in our experiments

Test dataset	Training dataset	k-GeneTriple	k-TSP
Singh	Welsh	5	3
	LaTulippe	5	3
	Welsh + LaTulippe	5	1
Welsh	Singh	9	1
	LaTulippe	5	3
	Singh + LaTulippe	7	5
LaTulippe	Singh	9	1
	Welsh	5	3
	Singh + Welsh	9	9

Table 11
Colon cancer cDNA microarray data

Data name	Number of genes	Number of normal samples	Number of tumor samples	Total number of samples	Characteristics
A_batch_Paired	17104	131	131	262	A batch, Paired, Used as Training Data
A_batch_Unpaired	17104	0	86	86	A batch, Unpaired, Used as Training and Test Data
B_batch_Unpaired	17104	0	211	211	B batch, Unpaired, Used as Training and Test Data

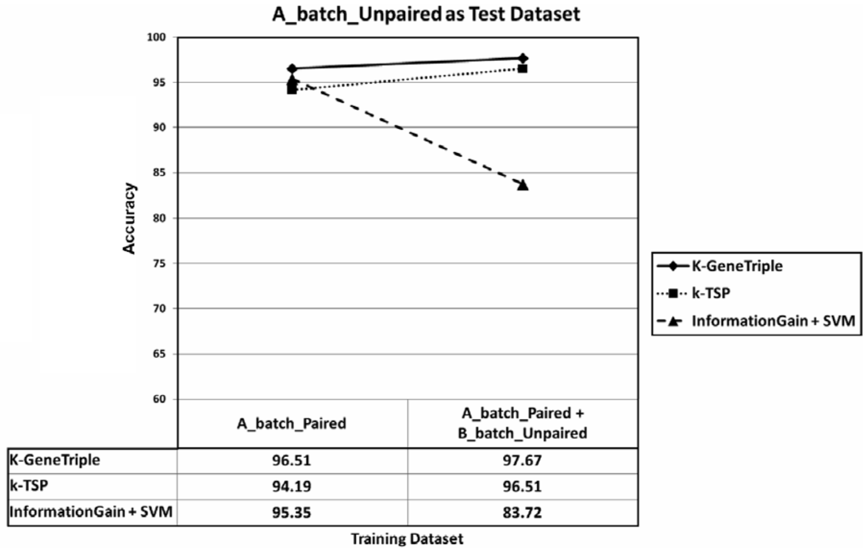
tissue sample were collected from the same origin (person), while "Unpaired" means that an individual tumor sample was collected from one person. The data was preprocessed with non-missing proportion (Non-Missing Proportion) of 100%.

As the previous subsection, we compared our system with other two methods: (1) *k*-TSP, (2) Information Gain + SVM. Table 12 shows the values of optimal *k* used in this experiments as a result of LOOCV run. As shown in Figs. 13 and 14, we built a classifier using a training dataset, which consists of all possible data combinations, excluding the test dataset, and measured the classification accuracies for each independent test data.

The classification accuracy rates for both A_batch_Unpaired and B_batch_Unpaired were similar and excellent as shown in Figs. 13 and 14. Using a rank value within a sample instead of an expression value took effect in eliminating variations between batches. Like Affymetrix data, the classification accuracy was higher as the sample size was larger by integration. The proposed two-stage approach can produce more credible classifiers than other two methods, especially when the data are integrated.

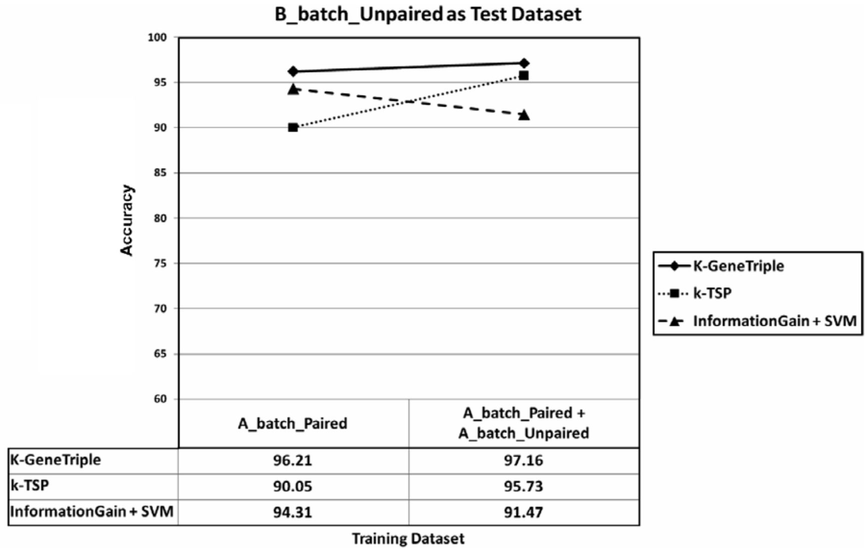
Table 12
The values of optimal *k* used in our experiments

Test data	Training data	*k*-GeneTriple	*k*-TSP
A_batch_Unpaired	A_batch_Paired	7	1
	A_batch_Paired + B_batch_Unpaired	7	9
B_batch_Unpaired	A_batch_Paired	7	1
	A_batch_Paired + A_batch_Unpaired	7	5



	A_batch_Paired	A_batch_Paired + B_batch_Unpaired
K-GeneTriple	96.51	97.67
k-TSP	94.19	96.51
InformationGain + SVM	95.35	83.72

Fig. 13. Accuracy when A_batch_Unpaired was used as test data.



	A_batch_Paired	A_batch_Paired + A_batch_Unpaired
K-GeneTriple	96.21	97.16
k-TSP	90.05	95.73
InformationGain + SVM	94.31	91.47

Fig. 14. Accuracy when B_batch_Unpaired was used as test data.

Since the test data does not include normal samples, the accuracy rate is same as the sensitivity, and specificity is not applicable.

5.4. Run-time comparison of *k*-GeneTriple and TSP

We compared time-complexity of *k*-GeneTriple and TSP in Table 13. In *k*-GeneTriple, the number of genes involved in classification stage is reduced by informative gene selection stage which selects 1% of informative genes out of original genes. Informative gene selection stage takes order of linearithmic time. The number of rules considered in classification stage is ${}_{n*0.01}P_3$ which is $(n * 10^{-2}) * (n * 10^{-2} - 1) * (n * 10^{-2} - 2)$ when the number of original genes is n. Since the number of informative genes is usually in the range of 50–200, it can be said that time complexity of classification stage of *k*-GeneTriple is O(1).

We made a run-time comparison of *k*-GeneTriple, and TSP. Table 14 presents the run-time for the Affymetrix datasets, and Table 15 presents the run-time for cDNA datasets. Experiments were conducted on a Pentium(R) 4 CPU 3.00 GHz PC with 1.00 GB RAM. For TSP we used the executables Tan [22] provides. Comparison of run-time reveals that *k*-GeneTriple runs 3.36 times faster than TSP at best, and runs 1.05 times

Table 13
Time complexity comparison of *k*-GeneTriple and TSP

Stage	*k*-GeneTriple		TSP	
	Time complexity	Comments	Time complexity	Comments
Informative gene selection stage	$O(p * n\log n + n * p\log p + n * p) = O(np\log np)$	n: number of genes, in tens of thousands p: number of samples, in hundreds $p * (n\log n)$: $p *$ (sorting a sample with n genes) $n * (p\log p)$: $n *$ (swap number counting for a gene) $n * p$: $n *$ (swap number counting for a gene)	None	
Classification stage	$O(C * n^3)$	${}_{n*0.01}P_3$: number of rules considered Only select 1% of informative genes C: 10^{-6}	$O(n^2)$	${}_nP_2$: number of rules considered
Total	$O(np\log np) + O(C * n^3)$	C: 10^{-6}	$O(n^2)$	

Table 14
Run-time comparison of *k*-GeneTriple and TSP for Affymetrix datasets in seconds

Test dataset	Training datasets	*k*-GeneTriple			TSP
		Informative gene selection	Classification	Total	
Singh	Welsh	3.422	156.641	160.063	242.156
	Latulippe	2.765	133.891	136.656	459.422
	Welsh + Latulippe	6.203	287.578	293.781	318.703
Welsh	Singh	9.516	442.469	451.985	511.218
	Latulippe	2.828	134.093	136.921	459.015
	Singh + Latulippe	11.953	561.75	573.703	614.797
Latulippe	Singh	9.656	442.171	451.827	510.515
	Welsh	3.672	156.75	160.422	241.047
	Singh + Welsh	12.937	584.703	597.64	628.968

Table 15
Run-time comparison of *k*-GeneTriple and TSP for cDNA datasets in seconds

Test dataset	Training datasets	*k*-GeneTriple			TSP
		Informative gene selection	Classification	Total	
A_batch_Unpaired	A_batch_Paired	16.313	332.11	348.423	510.641
	A_batch_Paired + B_batch_Unpaired	27.641	518.141	545.782	829.297
B_batch_Unpaired	A_batch_Paired	16.563	332.016	348.579	512
	A_batch_Paired + A_batch_Unpaired	20.438	379.047	399.485	594.031

faster than TSP at worst for Affymetrix datasets. For cDNA datasets *k*-GeneTriple runs 1.51 times faster than TSP at best, and runs 1.46 times faster than TSP at worst.

5.5. Effectiveness of the rank-based microarray data integration in classification

One of the problems in microarray data classification is that the number of genes far exceeds the number of tissue samples. We performed a direct integration of individual microarrays with same biological objectives by converting an expression value into a rank value within a sample, and applied a classification method with 100% of the original genes. We made all the rank-valued genes participate in building a classifier. What we want to demonstrate in this section is that bigger sample size by rank-based direct-integration with 100% of the original genes increases the classification accuracy when the typical classification method like SVM is used. SVM is the most common method for classification. We are able to show that the classification accuracy of an independent test dataset is getting higher as the sample size of the training dataset is bigger. A training dataset can be a single rank-valued microarray dataset or an integrated rank-valued microarray dataset.

Table 16
Classification accuracy of rank-valued Affymetrix test datasets using SVM

Rank-valued test dataset	Ranked-valued training datasets	Accuracy	Sample size
Singh	Latulippe	56.86	26
	Welsh	80.39	33
	Latulippe + Welsh	83.33	59
Welsh	Latulippe	75.76	26
	Singh	100	102
	Latulippe + Singh	100	128
Latulippe	Welsh	100	33
	Singh	100	102
	Welsh + Singh	100	135

Table 17
Classification accuracy of rank-valued cDNA test datasets using SVM

Rank-valued test dataset	Ranked-valued training datasets	Accuracy	Sample size
A_batch_Unpaired	A_batch_Paired	93.02	262
	A_batch_Paired + B_batch_Unpaired	95.35	473
B_batch_Unpaired	A_batch_Paired	91	262
	A_batch_Paired + A_batch_Unpaired	93.84	348

This rank-based direct integration method has an effect of enlarging the number of samples, and increasing the accuracy rate of the classification. Tables 16 and 17 show that the classification accuracy of rank-valued test dataset using rank-valued training datasets which consist of all possible data combinations, excluding the test dataset. One can see that the accuracy is getting higher as the sample size is bigger by integration. This rank-based direct integration with SVM is an effective method in microarray data classification. We were able to maximally use the abundant microarray data that is being stockpiled by the thousands of different research groups while improving classification accuracy.

6. Conclusion

In this paper, we introduce a novel two-stage approach for phenotype classification that sequentially combines independent microarray dataset integration, informative gene selection, and classifier building. Using the abundant supply of publicly available microarray data, we utilized a new method of integrating independently-generated microarray data with the same experimental objectives to select informative genes. We have discovered a more reliable classifier by increasing sample size through integrating independent microarray datasets. Moreover, two-stage approach made the computation time of the second stage tremendously lessen because only the pre-selected informative genes were considered. Since the number of genes involved in our classifier is relatively small, focusing on these genes could be very cost-effective in a clinical setting while microarrays with thousands of genes are impractical. Furthermore, our classifier has a straightforward biological interpretation. A prototype was implemented and tested on integrated prostate microarray datasets and colon cDNA microarray datasets. Our experiments show that our informative gene selection method is better than or comparable to other methods. Compared to the Information Gain plus the SVM method, our system had better classification accuracy with independent test datasets as the sample size of the training dataset grew larger by integration. We are currently investigating (1) cross platform validation where cDNA and Affymetrix data are included in the integrated microarray data, (2) elimination of redundancy among informative genes, and (3) generalization of the number of rules in the classifier and the number of genes involved in each rule.

Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. R01-2006-000-11106-0).

References

[1] N. Bailey, Statistical Methods in Biology, Cambridge University Press, Cambridge, United Kingdom, 1995.

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

[3] C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, New York, 1995.

[4] J.K. Choi, U. Yu, S. Kim, O.J. Yoo, Combining multiple microarray studies and modeling interstudy variation, Bioinformatics 19 (2003) 84–90.

[5] B. Dasarathy, Nearest Neighbor Norms: NN Pattern Classification Techniques, IEEE Computer Society Press, Los Alamitos, CA, 1991.

[6] S. Dudoit, J. Fridlyand, Classification in microarray experiments, in: T.P. Speed (Ed.), Statistical Analysis of Gene Expression Microarray Data, Chapman & Hall/CRC, 2003.

[7] U. Fayyad, K. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, in: Proceedings of the 13th International Joint Conference on Artificial Intelligence, Portland, OR, 1993, pp. 1022–1027.

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

[9] J. Han, M. Kamber, Data Mining: Concepts and Techniques, second ed., Morgan Kaufman, San Francisco, CA, 2006.

[10] H. Jiang, Y. Deng, H.S. Chen, L. Tao, Q. Sha, J. Chen, Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes, BMC Bioinformatics 5 (2004) 81–92.

[11] B. Jin, Y.C. Tang, Y.Q. Zhang, Support vector machines with genetic fuzzy feature transformation for biomedical data classification, Information Sciences 177 (2007) 476–489.

[12] T. Joachims, Dept. of Computer Science, Cornell University, SVM light <http://svmlight.joachims.org/>.

[13] J. Kang, J. Yang, W. Xu, P. Chopra, Integrating heterogeneous microarray data sources using correlation signatures, in: B. Ludäscher, L. Raschid (Eds.), Data Integration in the Life Sciences, Second International Workshop, DILS 2005, San Diego, CA, USA, 2005, pp. 105–120.

[14] E. LaTulippe, J. Satagopan, A. Smith, H. Scher, P. Scardino, V. Reuter, Comprehensive gene expression analysis of prostate cancer reveals distinct transcriptional programs associated with metastatic disease, Cancer Research 62 (2002) 4499–4506.

[15] L. Li, W. Leping, C.R. Weinberg, T.A. Darden, L.G. Pedersen, Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method, Bioinformatics 17 (2001) 1131–1142.

[16] T. Li, C. Zhang, M. Ogihara, A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression, Bioinformatics 20 (2004) 2429–2437.

[17] D.R. Rhodes, T.R. Barrette, M.A. Rubin, D. Ghosh, A.M. Chinnaiyan, Meta-analysis of microarrays: interstudy validation of gene expression profiles reveals pathway dysregulation in prostate Cancer, Cancer Research 62 (2002) 4427–4433.

[18] P.J. Park, M. Pagano, M. Bonetti, A nonparametric scoring algorithm for identifying informative genes from microarray data, in: Pacific Symposium on Biocomputing, Hawaii, 2001, pp. 52–63.

[19] Y. Peng, A novel ensemble machine learning for robust microarray data classification, Computers in Biology and Medicine 36 (2006) 553–573.

[20] M. Robnik-Sikonja, I. Kononenko, Theoretical and empirical analysis of relieff and rrelieff, Machine Learning 53 (2003) 23–69.

[21] D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, J. Manola, C. Ladd, Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2002) 203–209.

[22] A. Tan, D. Naiman, L. Xu, R. Winslow, D. Geman, Simple decision rules for classifying human cancers from gene expression profiles, Bioinformatics 21 (2005) 3896–3904.

[23] C. Tang, A. Zhang, J. Pei, Mining Phenotypes and Informative Genes from Gene Expression Data, in: ACM SIGKDD, Washington DC, 2003, pp. 24–27.

[24] K. Torkkola, R.M. Gardner, T. Kaysser-Kranich, C. Ma, Self-organizing maps in mining gene expression data, Information Sciences 139 (2001) 79–96.

[25] V. Vapnik, Statistical Learning Theory, John Wiley & Sons, New York, 1999.

[26] J.B. Welsh, L.M. Sapinoso, A.I. Su, S.G. Kern, J. Wang-Rodriguez, C.A. Moskaluk, Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer, Cancer Research 61 (2001) 5974–5978.

[27] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kaufman, San Francisco, 2005.

[28] Yonsei University, Cancer Metastasis Research Center, Yonsei University College of Medicine, South Korea.