THESIS

GENE SELECTION BASED ON CONSISTENCY MODELLING, ALGORITHMS AND APPLICATIONS

Submitted by

Yingjie Hu

The Knowledge Engineering and Discovery Research Institute

(KEDRI)

In partial fulfillment of the requirements

for the degree of Master of Computer and Information Sciences

Auckland University of Technology

July 2006

Abstract

Consistency modeling for gene selection is a new topic emerging from recent cancer bioinformatics research. The result of classification or clustering on a training set was often found very different from the same operations on a testing set. Here, the issue is addressed as a consistency problem. In practice, the inconsistency of microarray datasets prevents many typical gene selection methods working properly for cancer diagnosis and prognosis. In an attempt to deal with this problem, a new concept of performance-based consistency is proposed in this thesis.

An interesting finding in our previous experiments is that by using a proper set of informative genes, we significantly improved the consistency characteristic of microarray data. Therefore, how to select genes in terms of consistency modelling becomes an interesting topic. Many previously published gene selection methods perform well in the cancer diagnosis domain, but questions are raised because of the irreproducibility of experimental results. Motivated by this, two new gene selection methods based on the proposed performance-based consistency concept, GAGSc (Genetic Algorithm Gene Selection method in terms of consistency) and LOOLSc (Leave-one-out Least-Square bound method with consistency measurement) were developed in this study with the purpose of identifying a set of informative genes for achieving replicable results of microarray data analysis.

The proposed consistency concept was investigated on eight benchmark microarray and proteomic datasets. The experimental results show that the different microarray datasets have different consistency characteristics, and that better consistency can lead to an unbiased and reproducible outcome with good disease prediction accuracy.

As an implementation of the proposed performance-based consistency, GAGSc and LOOLSc are capable of providing a small set of informative genes. Comparing with those traditional gene selection methods without using consistency measurement, GAGSc and LOOLSc can provide more accurate classification results. More importantly, GAGSc and LOOLSc have demonstrated that gene selection, with the proposed consistency measurement, is able to enhance the reproducibility in microarray diagnosis experiments.

Acknowledgements

I wish to thank my supervisors, Dr. Paul Shaoning Pang and Dr. Ilkka Havukkala, for their guidance and support through this study. Dr. Paul Pang is my primary and technical advisor who helped me understand gene selection from the machine learning perspective. I owe a great deal to him for contributing the idea of new gene selection algorithm. It has been a great pleasure to have worked under his supervision over the last year. His instructions and comments were always essential to my study and ensured the research advancing in the correct direction.

Thanks also to Dr. Ilkka Havukkala for his advice and encouragement. He has always provided insightful perspectives toward this research and many papers in literature that are very helpful in the development of gene selection algorithms. His comments were essential in guiding this research for avoiding the generalization error in experiments. He also helped me to improve my English and carefully read many drafts of my paper.

At KEDRI, I would like to thank Prof. Nikola Kasabov, Peter Hwang, Joyce D'Mello and other staff who were always helpful in solving various problems for me. I also wish to thank Dr. Defoin Platel Michael for his advice and suggestion about optimization algorithms.

Finally, I am infinitely grateful to my wife, Sally Shen for her patience and encouragement throughout my study. Thanks also to my family for all they have done in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DA	-	Data-Adaptive
FDR	-	False discovery rate
GA	-	Genetic algorithm
GAGSc	-	GA gene selection method in terms of consistency
KNN	-	K-Nearest Neighbor
LOOE	-	Leave-one-out error
LOOLS	-	Leave-one-out Least-Square bound
LOOLSc	-	Leave-one-out Least-Square bound with Consistency measurement
LOOSVM	-	Leave-one-out SVM
LS-SVM	-	Least-square SVM
PCA	-	Principal Component Analysis
SAM	-	Significance Analysis of Microarrays
SBE	-	Sequential backward elimination
SFFS	-	Sequential floating forward selection
SFS	-	Sequential forward selection
SNR	-	Signal-to-Noise-Ratio
SVM	-	Support Vector Machine

LIST OF SYMBOLS

B	-	The number of resampling times for computing consistency
C	-	Consistency
D	-	Microarray data
D_a	-	Sampling distribution *a* from microarray data *D*
D_b	-	Sampling distribution *b* from microarray data *D*
F	-	Base function (e.g. F_t)
F_{sc}	-	The function of computing consistency under the condition of gene selection
F_{sp}	-	The function of computing performance under the condition of gene selection
F_t	-	Testing statistical function
f_s	-	Gene selection function
g_i	-	The ith gene in *D*
N	-	The number of generations in GA
P_a	-	Classification performance on D_a
P_b	-	Classification performance on D_b
S	-	The set of selected informative genes on dataset *D*
s_i	-	The ith selected gene in *S*
R	-	The ratio to consistency and performance
ρ	-	The number of initial selected genes
σ_i	-	Pooled standard deviation of the ith gene
σ_{yi}	-	The standard deviation of the ith gene corresponding to one class (e.g. class 0)
σ_{xi}	-	The standard deviation of the ith gene corresponding to another class (e.g. class 1)

Chapter 1

Introduction

1.1 Background

1.1.1 Recent research in microarray area

The advent of microarray technology has made it possible to monitor the expression levels for thousands of genes simultaneously, which can help clinical decision making in complex disease diagnosis and prognosis, especially for cancer classification, and for predicting the clinical outcomes in response to cancer treatment. It has been reported that the results of microarray experiments can be nearly 100% accurate (Petricoin, Ardekani et al., 2002; Zhu, Wang et al., 2003). Microarray technology is thus considered as a revolution for studying all human diseases, and is very important for developing complex diseases therapy schemes (Schena, 2002).

Microarray technology is capable of profiling differential gene expressions of tissue samples. Dozens of microarray research papers have shown that this technology is highly sensitive and specific to detect cancer and predict prognosis. The proponents of microarray technology even claim that “all human illness can be studied by microarray analysis, and the ultimate goal of this work is to develop effective treatments of cures for every human disease by 2050” (Schena, 2002).

The majority of current medical microarray research is conducted in the realm of cancer (or tumour) classification. Cancer diagnosis primarily relies on the histological appearances of the tumours, which has been proved not reliable and accurate. Moreover, during the treatment period of morphologically similar tumours, there are often different disease progressions and responses. Ideally, a systematic and unbiased method is able to successfully classify cancers. Hence, microarray technology has been put forward as a new aid in treating various cancers and related complex diseases.

The applications of microarray technology are able to utilize information and knowledge from human genome project to benefit human health. In the last few years, the remarkable progress achieved in microarray technology domain has helped researchers to develop the optimized treatment of cancer and other complex diseases, as well as the evaluation of prognosis based on genetic knowledge. For example, cDNA microarray is used to assess Parkinson's disease samples and examine the drug intervention (Mandel, Weinreb et al., 2003). Array technology has been employed in several studies of Alzheimer disease to predict different stages, including preclinical and prognosis stages (Galvin & Ginsberg 2004; Galvin, Powlishta et al., 2005).

1.1.2 Reproducibility and reliability problems due to Bias

However, whether microarray data analysis can accurately predict the cancer patients has been disputed in recent scientific literature, because many impressive results of microarray experiments could not be reproduced. For example, previous works on a well-known benchmark microarray dataset, ovarian data have been found to be irreproducible. In 2002 and 2003, two studies (Petricoin et al., 2002; Zhu et al., 2003) reported that the prognosis of ovarian cancer through microarray data technology was highly accurate (nearly 100% accurate). The outcomes were questioned when these testing approaches were intended to be commercialized, because the previous results were unrepeatable in practice (Wagner, 2004). Baggerly et al. (2005) refuted the results of these two major ovarian cancer studies, as their analysis showed that the classification of these microarray datasets was variable and the high prediction accuracies were unreachable due to overfitting caused by the unique structure of microarray data.

Reproducibility has been often criticized as one of the most important bias problems in microarray research. Recently, bias has attracted a lot of attention. Ransohoff (2005) pointed out that bias was a big threat to the validity of microarray data analysis, because most impressive results of microarray studies could not be reproduced in later simulating experiments. Ntzani and Ioannidis (2003) reviewed all microarray studies of cancer diagnosis that were accessible from MEDLINE (1995 ~ April, 2003), and they found that only 16% studies could be "subsequently replicated with formal statistical significance, without heterogeneity or bias" (Ioannidis, Trikalinos et al., 2003).

More of such concerns have been raised in recent years, and microarray technology is even argued as a "noise discovery". Marshall (2004) disputes the reliability of the outcomes of microarray experiments: "Thousands of papers have reported results obtained using gene arrays, … But are these results reproducible?". Furthermore, it is claimed that microarray technology applied in five out of seven studies performs no better than flipping a coin (Ioannidis, 2005; Michiels, Koscielny et al., 2005).

Thus, bias becomes a big concern in microarray data analysis, and can be a threat to the advent of microarray technology. It is critically important to find a scheme to determine whether the result is reliable and accurate. Otherwise, hundreds of microarray experiments will lead to waste of time and resources, without getting expected results. Recently, the academic community has recognized that it is crucial to establish a set of evaluation criteria that enables researchers to choose proper methodologies leading to more efficient and reliable outcomes. Consequently, plenty of literature has been published focusing on various techniques, such as estimating bias error, validation schemes and resampling methods (Allison, Cui et al., 2006; Braga-Neto, Hashimoto et al., 2004; Liotta, Lowenthal et al., 2005; Varma & Simon, 2006). Allison and his colleagues (2006) summarized several issues in microarray experiments, and suggested a verification system as well.

1.1.3 Microarray dataset structure: High dimensionality problem

Why microarray experiments are difficult to replicate? Empirical research has revealed that the main reason is the extremely unbalanced structure of microarray datasets (Chuang, Liu et al., 2004; Li & Yang, 2002; Pawitan, Murthy et al., 2005). In a typical microarray dataset, each row represents a tissue sample, and each column represents one gene. Because using the microarray chip is still expensive, the number of samples analysed is too small comparing to the number of the genes on the chip. In most real microarray datasets, the number of genes (usually thousands or tens of thousands) far exceeds the number of samples (usually tens or hundreds). For example, there are 78 samples vs. 24,482 genes in the breast cancer dataset of (van't Veer, Dai et al., 2002).

In machine learning research, in order to get a satisfactory classifying accuracy, the sample size of the dataset should be sufficiently large comparing to the number of features (Ambroise & McLachlan, 2002; Glymour, Madigan et al., 1996; Hosking, Pednault et al., 1997; Varma et al., 2006). Raudys (1976) indicates that a good classifier comes from a dataset with balanced structure, i.e. the sample size should be appropriate to the number of features. Generally speaking, the generalization error in machine learning area decreases when the sample size increases (Hamamoto, Uchimura et al., 1996).

However, it is not feasible to get a microarray dataset with a larger sample size, compared to the features (genes). This leads microarray analysis to become a formidable challenge. A huge number of genes usually include many redundant genes (noise genes) that can confuse a classifying algorithm. Moreover, the huge dimensionality problem makes microarray data analysis very costly in terms of time and computation.

Therefore, a major challenge with microarray research is how to find informative genes that can be used for effective discriminating variables in relation to different conditions, such as classifying healthy and diseased tissue samples. The amount of relevant genes is typically small, as “the majority of the active cellular mRNA is not affected by the biological differences”(Wolf, Shashua et al., 2004). Previous classification work on microarray datasets have shown that using a small number of informative genes can successfully discriminate the tissue sample types, e.g. diseased or healthy (Dudoit, Fridlyand et al., 2000; Eisen, Spellman et al., 1998; Golub, Slonim et al., 1999).

1.1.4 Gene selection

Gene selection is not a brand-new technology, in terms of the technological aspect. In the viewpoint of data mining, gene selection can be seen the feature selection which is widely used in data pre-processing stage. However, gene selection, unlike feature selection in the area of machine learning, is characterised by the great difference between a huge number of genes (usually thousands or tens of thousands) and a very small number of samples (typically tens).

There are plenty of reasons for employing gene selection, especially in cancer diagnosis and treatment area. The main benefits of employing gene selection methods in microarray data analysis can be summarised as follows:

a). The cost of cancer diagnosis in terms of time and computation can be greatly reduced. It is much cheaper to focus on a small number of informative genes that can differentially express the patterns of disease from the whole gene set (Tang, 2006).

b). Most noise genes can be removed. As mentioned above, the presence of many noise genes is the main reason causing high generalization error. Therefore, the performance of microarray experiment will be improved if these noise genes are eliminated.

Therefore, using effective gene selection methods, a small list of highly informative genes can be discovered from whole gene set. Then, these genes can be utilized to construct the classifier for discriminating disease patterns.

1.1.5 Gene selection methods in literature

During the last few years, gene selection has become a hot topic that has attracted great attention in bioinformatics area, and a number of methods and algorithms have been published. Simple gene selection methods come from statistical methods, such as t-statistics, Fisher’s linear discriminate criterion and Principal Component Analysis (PCA) (Ding & Peng, 2003; Furey, Cristianini et al., 2000; Jaeger, Sengupta et al., 2003). They are usually effective and run very quickly. To improve the efficiency of selected genes in terms of disease prediction accuracy, more sophisticated algorithms have been proposed, e.g. Noise sampling method (Draghici, Kulaeva et al., 2003), Bayesian model (Efron, Tibshirani et al., 2001; Lee, Sha et al., 2003), and Significance Analysis of Microarrays (SAM) (Tibshirani, 2006). In addition, artificial neural networks are also being used for gene selection, a representative work in this category is evolving connectionist system (ECS) (Kasabov, Middlemiss et al., 2003). Most of above methods are claimed to be capable of extracting out a set of highly informative genes (Wolf et al., 2004).

Gene selection methods can be generally classified into two major groups: filter and wrapper methods, depending on whether the learning algorithm is used as a part of the selection criteria (Ambroise et al., 2002; Devijver & Kittler, 1982; Inza, Larranaga et al., 2004). Filter method examines the intrinsic characteristics of genes as the measuring criterion. The gene selection procedure is independent of the classification process, as the classifier is not constructed before informative genes are selected out. In contrast, wrapper method evaluates genes based on the performance of an induction algorithm usually involving a classifier. The selection algorithm is learnt and optimized during the gene selection process, and implemented to the gene evaluation criterion as well. In wrapper methods, it can be said that gene selection process is wrapped around a specific machine learning algorithm.

Filter method is more popular than wrapper method in gene selection area, because it can generally achieve satisfactory performance with much less computational cost. Filter gene selection methods can be found in many published works: A Noise sampling method based on an ANOVA approach (Draghici et al., 2003), minimum redundancy – maximum relevance (MRMR) gene selection method (Ding et al., 2003), Self Organizing Maps (SOM) based method (Tamayo & et al., 1999), Singular Value Decomposition (SVD) (Atler & et al., 2000), a.k.a gene shaving method (Hastie & et al., 2000), max-surprise method (Ben-Dor, Friedman et al., 2001), and so on.

In most pattern recognition applications, wrapper method outperforms filter method. However, the better performance obtained from wrapper method is coupled with high cost in terms of time and computational complexity, which is reported by the authors in several papers (Guyon & Elisseeff, 2003; Kohavi & John, 1997). In wrapper method, the gene selection process is heavily dependent on a search engine, a search area (data), and an evaluation criterion to optimize the gene selection approach. A simple flow structure of wrapper method is shown in Fig. 1.1.

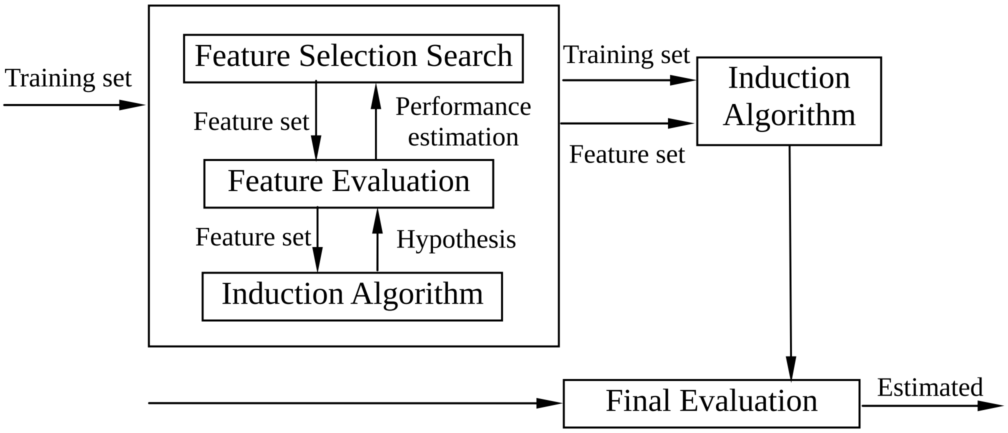


Fig.1.1 A simple flow structure of wrapper method (adapted from Kohavi et al., 1997)

Wrapper method has been widely accepted in gene selection since it was proposed by Kohavi (1995) in his feature selection work, in which wrapper method was shown more efficient than filter method with respect to the classification accuracy. However, unlike the high popularity of filter gene selection methods employed in microarray research, there are relatively few works based on wrapper methods because of its cost-ineffectiveness. Guyon et al. (2002) used the wrapper method consisting of a Support Vector Machine (SVM) algorithm based on Recursive Feature Elimination (RFE) to select informative genes for leukaemia and colon cancer data. Li and Xiong (2002) used the wrapper approach for a Fisher's linear discriminate algorithm, and showed it very sensitive for gene selection (with only 6 genes, the classification accuracy on colon cancer data can be over 90%). Lee et al. (2003) developed wrapper gene selection approach in the context of a hierarchical Bayesian model with Markov Chain Monte Carlo (MCMC) search algorithm for finding informative genes.

1.2 Motivation

1.2.1 Consistency issue in gene selection

In our previous experiments, the results obtained from the operation, such as classification and clustering on the training dataset were found very different from that of the same operations on the testing dataset. For example, the training set from CNS cancer data (Pomeroy, Tamayo et al., 2002) can get a performance of above 90% true positive (TP) accuracy for tumour classification, whereas the testing data only gets 70% of TP accuracy. This occurs because those typical methods for gene selection use only a single criterion of distance measurement between patients and non-patients, but regardless of the consistency between the subsets of data with the genes selected under the criterion.

In this thesis, this issue is discussed, which is here referred to as the consistency problem. Moreover, it is also noticed that selecting a set of proper genes can significantly reduce the inconsistency of microarray data experiment. Obviously, it will be more interesting to find out a set of genes that enable a consistently good classification performance over different subsets of patients in the complete microarray data.

1.2.2 Related work on consistency issue in gene selection

The concept of consistency is proposed for the purpose of improving the reproducibility of gene selection in microarray experiments. Since the importance of consistency for microarray data analysis has not been sufficiently recognised by bioinformatics researchers so far, there are very few published works related to this new topic. In addition, there is no official definition of the concept of consistency used in microarray research, and it is defined in different ways which will be described in later sections.

Probabilistic consistency analysis for gene selection method (Mukherjee & Roberts, 2004) is a recent novel approach that focuses on analyzing the common genes selected from two datasets (Mukherjee, Roberts et al., 2005). The consistency is defined as the number of genes in common between two gene sets. This probabilistic consistency concept is applied to their gene selection method for selecting truly differentially expressed genes under various conditions. In the process of gene selection, the result of consistency computed from top-ranked genes is used for optimizing a test statistics function. After hundreds of iterations, an optimized statistic function can be achieved based on the consistency improvement. Then, a small list of most informative genes can be discovered with the final optimized statistic function.

In short, the concept of consistency proposed in their paper is based on the number of genes in common between two subsets. The value of consistency is dependent on several factors, including ranking function, number of selected genes, and iteration times. The authors indicate that the more informative the genes selected, the better consistency the experimental results. Their experiment shows the gene selection methods with consistency concept is effective for improving the reproducibility of microarray analysis (Mukherjee et al., 2005). Their method is described in detail with a simulation experiment in chapter 2.

1.2.3 Gene selection method in terms of consistency

The concept of consistency proposed by Mukherjee et al. (2005) focuses on the common genes selected from two sampled datasets. However, it is not clear to what extent the selected “highly differentially expressed genes” in terms of consistency are related to the performance of the classification or clustering on microarray data. In other words, the performance of classification or clustering over a dataset may not be improved significantly, though the method based on their consistency concept is employed. Motivated by this issue, a new consistency concept in terms of performance is proposed in this thesis.

The idea of the new gene selection methods proposed in this thesis is to use the result of consistency obtained from an operation (e.g. classification or clustering) to find informative genes for a microarray dataset. For most microarray datasets, there tends to be no agreement on which genes are highly differentially expressed, and consequently it is difficult to measure the reliability of any gene selection method. In practice, the performance of an operation over microarray data is a straightforward criterion for measuring the outcomes of microarray experiments. The proposed solution is based on the optimizing computation that takes consistency measurement into account.

1.3 Organization of the thesis

This study is organized into the following chapters:

Chapter 2 provides a literature review of several widely-used gene selection methods, including t-test, SNR, SAM, and the Data-Adaptive (DA) method with the consistency of common genes (Mukherjee et al, 2005).

Chapter 3 presents a definition of the proposed performance-based consistency concept, and describes two proposed consistency-based gene selection methods, GAGSc (Genetic algorithm gene selection method in terms of consistency), LOOLSc (Leave-One-Out Least-Square bound method with consistency measurement) and relevant algorithms.

Chapter 4 presents the experimental results obtained by two proposed gene selection methods (GAGSc and LOOLSc) on seven benchmark microarray datasets and one proteomics dataset. The classification results from GAGSc method is compared with the reported classification performance from the literature of microarray data analysis. The efficiency of the proposed performance-based consistency concept is examined by the comparison of LOOLSc and LOOLS (with consistency measurement vs. without consistency measurement). This chapter also discusses a totally unbiased validation scheme used in this study.

Finally, chapter 5 contains the discussion and conclusions of this study as well as suggestions of future work.

Chapter 2

Literature review: Gene selection methods

This chapter examines several commonly used approaches for gene selection in microarray studies. Numerous gene selection methods have been proposed in the literature, as gene selection is considered one of the main tasks for microarray research. To find informative genes, a diversity of techniques and approaches derived from different areas, such as statistical theory, neural network and genetic algorithms have been introduced to microarray studies. Three popular algorithms used in gene selection methods are reviewed in this chapter, including T-test and its varieties, Significance Analysis of Microarrays (SAM) and Signal-to-Noise-Ratio (SNR). Additionally, one recently published gene selection method (Mukherjee et al., 2005) based on a consistency concept is discussed in detail.

Gene selection has been found useful for improving the consistency in terms of the performance for the classification on microarray data. Here, the concept of consistency is defined as the absolute difference between the performance on a training set and on a testing set. Consider a microarray dataset having two predictor classes, e.g. healthy or diseased. Using cross validation, the result obtained from a training set has been often reported to be very different from that from a testing set (Jain, Duin et al., 2000). This means the consistency of this microarray dataset is fairly low, so that the experimental results may vary quite significantly.

The aim of gene selection is to find a small group of informative genes that can successfully classify any samples from randomly resampled dataset into correct classes, and give consistently good results. For example, if the 5-fold cross validation technique is used in the experiment, the accuracy of classification on the 1 fold testing set should be very similar to that on the 4 folds training set. These selected genes thus can be regarded as the informative genes in terms of good consistency.

2.1 T-test based gene selection methods

2.1.1 Overview of T-test algorithm

T-test, since first published by Gosset (1908), has been extensively studied in the realm of machine learning and bioinformatics. T-test, as a classical statistical theory, is commonly applied to the judgements for measuring the differences in means between two distributions of a dataset. Theoretically, the T-test can perform well even if the number of samples is very small (Triola, 1998). This characteristic has made T-test widely used for gene selection in microarray research (Arfin, Long et al., 2000; Ding et al., 2003; Tanaka, Jaradat et al., 2000; Thomas, Olson et al., 2001). In practice, many previous microarray studies have shown that T-test or its varieties, e.g. Wilcoxon rank (Wilcoxon, 1945), and Westfall-Young (Westfall & Young, 1993) algorithms are effective for identifying differentially expressed genes for microarray studies (Ding, 2002; Dudoit et al., 2000; Model, Adorján et al., 2001). Using T-test algorithm, a small number of informative genes can be identified based on their intrinsic characteristics in relation with the target class labels.

Generally, the main idea of using T-test algorithm in gene selection is to evaluate to what extent each gene in a sample is related with a particular gene in other samples. The expression level of each gene is evaluated by t-test statistic. Suppose a two-class microarray dataset *D* pertaining to a gene selection task, and the t-test statistic value of each gene in *D* can be computed by:

$$T_i = \frac{\overline{X_i} - \overline{Y_i}}{\sqrt{(\frac{1}{n_x} + \frac{1}{n_y})\,\sigma_i}} \tag{2.1}$$

where T_i is the T-test statistic value of i^th^ gene in *D*, $\overline{X_i}$ and $\overline{Y_i}$ represent the mean value of i^th^ gene corresponding to different classes (e.g. class 0 and class 1) respectively. n_x and n_y are the number of samples of two classes (class 0 and class 1). σ_i is the pooled standard deviation for the i^th^ gene:

$$\sigma_i = \sqrt{\frac{(n_x - 1)\sigma_x^2 + (n_y - 1)\sigma_y^2}{df}} \tag{2.2}$$

where σ_x^2 and σ_y^2 are the variance of two subsets corresponding to different classes respectively.

$$\sigma_x^2 = \frac{\sum_{j=1}^{n_x}(X_j - \overline{X})^2}{n_x - 1} \tag{2.3}$$

$$\sigma_y^2 = \frac{\sum_{j=1}^{n_y}(Y_j - \overline{Y})^2}{n_y - 1} \tag{2.4}$$

df is the degrees of freedom of the t-distribution under the null hypothesis and calculated by:

$$df = (n_x + n_y - 2) \tag{2.5}$$

Thus, all genes can be ranked according to the scores of their T-test statistic during the gene selection process, so that a small number of high-ranked genes can be selected consequently.

One should bear in mind is that T-test distribution can be used only when the data is normally distributed and the population variances are equal in two classes. If variances are unequal in terms of two classes, the degrees of freedom (*df*) is computed by a different version of T-test, Welch's T-test (Welch, 1938) . The value of degrees of freedom obtained by Welch’s T-test is normally smaller than that obtained in Equation (2.5). Other different versions of t-test are also used depending on special conditions, for example, Levene's test (Levene, 1960) and Bartlett's test (Snedecor & Cochran, 1989) are two sensitive methods when the samples have equal variances (homogeneity of variances) (Snedecor et al., 1989). Note that parametric tests may perform poorly due to violation of their underlying assumptions, such as normality and equal variance in different groups (Hwang, 2002).

2.1.2 Applications of T-test and variants in gene selection

The classical T-test algorithm is probably one of the most popular techniques used for identifying significant difference between two sets of normalized data. For example, the two-sample T-test is a simple approach, and often applied to gene selection problems. Two-sample t-test takes the assumption that samples are randomly selected from normally distributed samples with equal variances. This algorithm was used in the gene selection method proposed by Dudoit et al. (2002), in which the differentially expressed genes were evaluated by the T-statistic value of Equation (2.1). The absolute expression level of the ith gene is further measured by $\overline{l_i}$:

$$\overline{l_i} = \frac{\sum_{j=1}^{n} log_2\sqrt{RG}}{n} \tag{2.7}$$

where *R* and *G* represent the intensity measurements for each gene spotted in a single-slide cDNA microarray chip, *n* is the number of hybridizations performed.

The more precise expression levels of genes can be measured by calculating p-values for each gene. However, as a typical microarray dataset consists of thousands of genes, the multiple testing becomes a big concern (Holm, 1979; Shaffler, 1986; Westfall et al., 1993). When the tests are performed many times, the probability of at least one Type I Error would be significantly increased. To deal with issue, several approaches are suggested to control this type of error, e.g. adjusted p-values (Shaffer., 1995), Bonferroni method (Bonferroni, 1936) and Westfall-Young step-down method (Westfall et al., 1993). Suppose p_i and $\widetilde{p_i}$ represent the unadjusted and adjusted p-values of the ith gene reprehensively. The Bonferroni single-step adjusted p-values are defined by:

$$\widetilde{p}_i = \min(np_i, 1) \tag{2.8}$$

where, n is the number of genes in the dataset.

In terms of unique characteristics of microarray dataset (i.e. the great difference between numbers of genes and samples), Westfall-Young step-down adjusted p-values are considered to be more general and accurate (Dudoit, Yang et al., 2002) in the measurement of the statistic value of the ith gene. The Westfall-Young p-value of the ith gene is calculated as follows (Dudoit et al., 2002):

$$\widetilde{p_1} = \text{pr}(\min p_i \leq p_1 \mid H_0) \mid i \in \{1, \ldots n\}$$
$$\widetilde{p_i} = \max(\widetilde{p}_{i-1}, pr(\min p_i \leq p_1 \mid H_0) \mid i \in \{1, \ldots n\} \tag{2.9}$$

where H_0 is the intersection of all null hypotheses, and p_i represents the unadjusted p-value of the ith gene. Let T_i denotes the t-test statistic value of the ith gene. Then, the permutation p-values for the T-test of ith gene in Dudoit's experiments are given by:

$$p_i^* = \frac{\sum_{j=1}^{m} \mathbf{I}(|T_i^{(j)}| \geq |T_i|)}{m} \tag{2.10}$$

where m is the number of iterations, and **I** is an indicator function for indicating the condition in parentheses, i.e. if the condition is true, then **I** returns 1, otherwise 0. In their two experiments, by using T-test methods with adjusted p-value approach, a small group of genes are selected out according to their statistic ranking scores. The genes found by this method seemed efficient in explaining the different patterns of two groups of mice models (Dudoit et al., 2002).

Other sophisticated gene selection methods based on classical T-test theory have been put forward to explore the genes whose expression patterns are highly correlated with the predictor classes. The statistic score of each gene can be obtained from a variation of classical T-test statistics algorithm proposed by Golub and his colleagues (Golub et al., 1999). Their method is called Neighbourhood analysis in which each gene is denoted by an expression vector:

$$v_{(i)} = (l_1, l_2, \ldots l_n) \tag{2.11}$$

where l_i represents the expression levels of the ith gene in *n* samples, and *n* is the number of samples in the given dataset *D*. According to the sample belonging to each class (class 1 or class 0), c is assigned to denote the expression patterns of class distinction. Then, they give the following formula to measure the correlation between a gene and a class distinction:

$$T_{(i)} = \frac{\overline{X_i} - \overline{Y_i}}{\sigma_{xi} + \sigma_{yi}} \tag{2.12}$$

where σ_{xi} and σ_{yi} are the standard deviation the ith gene corresponding to two different classes. The large value of $T_{(i)}$ reflects a strong correlation between the expression level of the ith gene and the class distinction.

Using neighbourhood analysis method, 50 genes were selected to do clustering over ALL-AML leukaemia dataset and the result obtained from the SOM (self-organizing maps) clustering reached very high accuracy, nearly 100% (Golub et al., 1999). Another contribution of their work is that they indicated with effective gene selection methods, it is feasible to predict cancer classes for other types of cancer without previous biological knowledge (Golub et al., 1999). However, recently these results have been criticized as unreplicable (see introduction).

The Wilcoxon rank test (Wilcoxon, 1945) is a non-parametric alternative to classical T-test statistics algorithm and has been reported powerful in the applications of gene selection (Guan & Zhao, 2005; Jaeger et al., 2003). The advantage of Wilcoxon rank test arises when the T-test statistic value calculated by Equation (2.1) is greater than the threshold specified for statistical significance, so that two observed samples cannot be discriminated according to standard T-test statistic criteria.

The Wilcoxon rank test for gene selection can be briefly described as follows:
Suppose a microarray dataset *D* consisting of *n* samples belonging to two different classes. X_i and Y_i represent the ith gene corresponding to two different classes. The Wilcoxon ranked statistic value for a gene is given by:

$$W_i = \sum_i^n \mathbf{I}(|X_i - Y_i|)R_i \tag{2.13}$$

where R_i is a ranking function, and **I** is an indicator function that indicates the conditions of parentheses.

2.1.3 Discussion of T-test gene selection method

T-test based algorithms are often used for comparing with other new developed gene selection methods, as they have been extensively studied and discussed in various research areas for decades. One of their major advantages is the simplicity and robustness, which leads to a fast computation process for gene selection. However, one must bear in mind that the high false-positive rate should be considered, when multiple tests are applied in the microarray data analysis. Otherwise, the differentially expressed genes cannot be discovered according to a typical P-value (0.05) that is commonly adopted for signifying differential expression levels for the genes responding to two groups (Ding, 2002).

In addition, T-test based gene selection algorithms usually make the assumptions that two samples have equal variances and the genes are independent. These assumptions can have a big negative impact on real microarray datasets. Empirical studies have indicated that the selected genes by simple T-test based algorithms are not reliable in terms of expressing disease patterns, and are easy to be generated by chance. For example, even if the P-value is significantly small (0.01) in a microarray experiment with 10,000 genes, 100 genes might be identified by chance. This issue has led scientists to develop more specific gene selection methods for microarray data analysis. SAM (Tusher, Tibshirani et al., 2001) is one of these methods and described in the following section .

2.2 SAM method

2.2.1 Overview of SAM algorithm

SAM is a recently proposed method specifically designed for gene selection in microarray data analysis. It is a statistical technique derived from T-test statistic for identifying informative genes in a set of microarray experiments. The method was first proposed by Tusher et al. (2001) and the software package was developed by Narasimhan at Stanford University. Empirical studies have shown that SAM method can get a good performance for gene selection, especially in small sample microarray dataset (Wu, 2005).

SAM method assigns a statistical score to each gene based on the change in gene expression in relation with the standard deviation computed from plenty of iterated measurements. Those genes whose scores greater than a pre-specified threshold are seen as potentially informative genes. To describe the SAM algorithm, certain basic concepts and definitions are needed to be explained.

FDR (false discovery rate), since first introduced by Benjamini and Hochberg (1995), has been extensively studied and used for controlling complicated errors in multiple-hypothesis testing. This quantity is the expected percentage of false positive findings among all the rejected hypotheses. In microarray studies, FDR is evaluated by a set of iterated measurements in which useless gene are identified. Then it is used for estimating the percentage of those genes identified by chance. During the process, the threshold for evaluation is adjustable to identify different size sets of genes, and a serious of FDR can be calculated regarding to each set.

FDR is defined as follows: with a set of m multiple tested null hypotheses, m_0 are true while m_1 are false. For each hypothesis H_i, a test statistic T_i is calculated for each gene along with the p-value. R is the number of hypotheses rejected by a procedure. V is the number of null true hypotheses rejected, which means the true positives among the selected genes. S denotes the number of false hypotheses rejected, i.e. the genes that are erroneously selected as informative. Thus, FDR is given by:

$$FDR = E\left(\frac{V}{R} \mid R > 0\right) \tag{2.14}$$

For simplicity, Table 2.1 summarizes the possible outcomes occurring when multiple hypothesis (m) tests are performed on a set of genes.

	Accept (not significant)	Reject (significant)	Total
Null True	U	V	m_0
Alternative True	T	S	m_1
Total	m - R	R	m

Table 2.1 Possible outcomes from m hypothesis tests of genes (adapted from Tibshirani, 2006)

SAM uses iterated permutation of the data to find the genes whether they can differentially expressed the pattern of samples. The procedure of SAM gene selection method starts with the evaluation of the standard deviation of each gene. The statistic value of each gene computed by SAM is derived from T-test statistic using Equation (2.1). Then, the correlation between different genes can be accounted for. According to a given hypothesized criterion, the false discovery rate (FDR) of a set of genes is evaluated. After a number of iterations, each gene has a statistic value that is capable of measuring the strength of the relationship between gene expression and the target variables (class labels) (Tusher et al., 2001).

Then SAM uses the following procedure to reduce the FDR occurring in the above statistic computation (Tibshirani, 2006):

1. Sort the statistic value in ascending order: $T_{(1)} \leq T_{(2)} \cdots \leq T_{(i)} \mid i = 1, 2, \cdots n$
2. Take permutations to a randomly chosen set of the statistic scores $T_{(\mathrm{i})}$ belonging to class 2 (here is $Y_{i=1,2\cdots m}$).
3. Adjust the small value of $T_{(\mathrm{i})}$ and intend to find a fixed threshold C.
4. Evaluate the FDR through the following criterion:

$$\bar{T} - T_{(i)} > C \cdot \tag{2.15}$$

The procedure runs *m* repetitions, and a small group of genes is finally selected into the informative gene list. The statistic value of each gene obtained from SAM algorithm is therefore defined as:

$$T_i = \frac{\overline{X_i} - \overline{Y_i}}{\sqrt{\frac{1}{n_x} + \frac{1}{n_y}} \left(\sqrt{\frac{(n_x - 1)\sigma_x^2 + (n_y - 1)\sigma_y^2}{df}} + C \right)} \tag{2.16}$$

where n_x and n_y are the number of samples responding to two classes, respectively.

2.2.2 Applications of SAM and its variants in microarray studies

SAM is a relatively new method for gene selection that makes gene selection more efficient and reduces the error rate simultaneously. Several recently published papers have applied SAM algorithm to the gene selection methods (Reiner, Yekutieli et al., 2003; Wu, 2005).

As already mentioned, T-test statistic is used in SAM method for calculating the statistic value of each gene. It is possible to use other techniques to make SAM method more robust. Motivated by this factor, other statistic algorithms have been accepted to take the place of T-test in SAM. Wu (2005) improved standard SAM with F-statistic and a penalized regression algorithm to evaluate gene expression levels. In addition, a linear regression models are used to do comparison for ith gene. Thus, the statistic value of the ith gene is calculated by:

$$T_i = \frac{\sum_{j \le n.} (X_{ij} - \bar{X}_i)^2 + \sum_{j > n.} (Y_{ij} - \bar{Y}_i)^2}{n-2} \tag{2.17}$$

where X_{ij}, Y_{ij} denote the value of the i^{th} gene corresponding to the j^{th} sample of class 1 and class 2, respectively, and n is the number of samples of whole dataset. Using F-statistic, the expression levels of genes are evaluated based on two groups of samples, one is for n_x samples from one class and n_y samples from another class.

The main strength of SAM method is the efficient control of FDR. In SAM method, FDR is a key factor used for improving the effectiveness for gene selection approach. Some varieties of SAM method with adjusted FDR definitions have been proposed (Hero, 2003; Reiner et al., 2003). For example, a variant of SAM method consists of evaluating the statistic ranking score of each gene by an enhanced FDR formula is proposed by Reiner et al (2003). In their method, for the p-values corresponding to the true null hypotheses, if the number of resampling-based p-values less than a specified threshold, they are denoted by $V_{(p)}$ that is an estimated upper bound. Then, FDR is calculated by:

$$FDR' = E\left(\frac{V_{(p)}}{V_{(p)} + \hat{S}_{(p)}} \right) \tag{2.18}$$

where $\hat{S}_{(p)}$ is denoted by the estimated number of false null hypotheses smaller than p.

SAM method is regarded as a very practical tool for finding informative genes with a satisfactory FDR level in gene selection. The multi-parametric assumptions about the distribution of individual genes can be avoided, because the statistic correlation between each gene and others in the dataset is accounted for numbers of distributing permutation. One disadvantage of SAM method is: in the permutation stage, all genes are put into one group for evaluation, which requires an expensive computation and probably confuses the analysis because of the noise genes.

2.3 SNR gene selection method

2.3.1 Overview of SNR algorithm

Another popular algorithm implemented in gene selection is signal-noise-to-ratio (SNR). SNR is often adopted for evaluating the expression level of each gene to conduct the search for an informative gene set. This approach starts with the evaluation of a single gene and iteratively searches the informative genes in the rest of dataset in terms of a statistic criterion. SNR, as a simple algorithm, is usually found generally effective to identify the difference between two normal distributed samples (Lai, Reinders et al., 2004; Veer, Dai et al., 2002).

Let $\overline{X_i}$ and $\overline{Y_i}$ denote the mean values of the ith gene for the samples in class 1 and class 2 respectively, σ_{xi} and σ_{yi} are the corresponding standard deviations. Therefore, the SNR score of each gene can be calculated by:

$$SNR_i = \frac{|\overline{X}_i - \overline{Y}_i|}{\sigma_{xi} + \sigma_{yi}} \mid i = 1, 2 \cdots n \qquad (2.19)$$

where n is the number of genes in the objective dataset D. With Equation (2.19), the greater the SNR value, the more informative related to the gene (Tibshirani, 2006).

2.3.2 Applications of SNR in gene selection

The implementation of SNR in gene selection can be found in the novel approaches in which other techniques and algorithms are combined. Such examples are the univariate ranking method (Lai et al., 2004) and a novel hybrid method (Goh, Song et al., 2004). For gene selection, SNR is usually employed to rank the correlated genes in the dataset based on their discriminative levels towards the classes. The genes with high SNR scores are chosen as the informative ones of each class.

One example of the application of SNR in gene selection is provided by Goh et al (2004), who proposed a hybrid method of Pearson correlation coefficient (PCC) and signal-to-noise ratio (SNR). The main idea of their method is using the hybrid method to measure the strength of correlation between two objectives: genes and classes. The linear correlation coefficient is measured by:

$$r_i = \sum (\frac{(g_i - \overline{g}_i)(y_i - \overline{y}_i)}{\sigma_{gi} + \sigma_{yi}} \mid i = 1, 2 \cdots n \qquad (2.20)$$

where:

g_i : the value of the ith gene in a test sample)

$\overline{g_i}$: the mean of the ith gene corresponding to all samples

y_i : the class of the ith gene in a test sample)

$\overline{y_i}$: the mean of the ith gene corresponding to all samples

σ_{gi} : the standard deviation of the ith gene

σ_{yi} : the standard deviation of the class corresponding to the ith gene

The linear correlation coefficient r_i is calculated by PCC and provides a mathematical linear dependency between g_i and y_i. However, it is found that PCC computation is very time-consuming, when the dataset becomes large.

To reduce the computational complexity, SNR is introduced into the hybrid method for measuring the importance of genes. All genes are calculated by Equation (2.20) and then each gene has its correlation coefficient (r_i) corresponding to other genes. The genes are selected if their r_i is greater than a given threshold. Then SNR is used to select a set of high-ranked genes to represent the corresponding groups of correlated genes. This operation can remove many noise genes, which significantly reduces the size of dataset for the next PCC calculation. The gene selection runs a number of iterations, and a group of genes with high SNR scores can be found for classification.

The selected genes are further measured by the result obtained from an evolving classification function (ECF). The importance of selected genes is evaluated according to the result of classifications. Finally, with this novel gene selection method, a list of genes regarded as the representatives carrying most important information of expression levels can be selected out. A limitation of this method is many genes with very low coefficient will be removed by the ranking criterion, because the correlation coefficient of genes is only measured by one gene to others. However, it is very likely that some of these abandoned genes are very useful for pattern express, when they are combined together for measuring the correlation.

In real clinical area, SNR is often used in the establishment of a diagnostic system to improve the therapeutic decision making and to reduce the unnecessary side effects of anticancer drugs. For example, docetaxel is one of the most effective anticancer drugs for breast cancer treatment. However, empirical studies have reported that nearly 50% of treated breast cancer patients do not have good response to it, but instead suffer many side effects (Iwao-Koizumi, Matoba et al., 2005). Here, these patients are addressed non-responders. Motivated by this issue, a weighted-voting (WV) algorithm combined with SNR method was proposed by Iwao-Koizumi et al. (2005) to evaluate the response of docetaxel.

In their method, the importance of each gene (vote value) of breast cancer data is evaluated by WV algorithm and denoted as V_i as follows:

$$V_i = W_i \times \left| X_i - \frac{\overline{X}_{ri} - \overline{X}_{ni}}{2} \right| \tag{2.21}$$

where, X_i is the expression level of the ith gene in a testing sample. X_{ri}, X_{na}, are the mean expression level of the ith gene in responders and the mean expression level in nonresponders respectively. W_i is the weight of the ith gene and calculated by SNR using Equation (2.19). Then, the prediction strength (PS) is denoted as the prediction of responders and nonresponders:

$$PS = \frac{V_r - |V_n|}{V_r + V_n} \tag{2.22}$$

where V_{r} represents the sum of positive votes of all diagnostic genes in responders, while V_{n} represents the sum of negative votes in nonresponders. The experiment showed that WV algorithm combined with SNR outperformed other algorithms in terms of classification accuracy. Fig. 2.1 shows the comparison of the result obtained by three methods based on classification on breast cancer data.

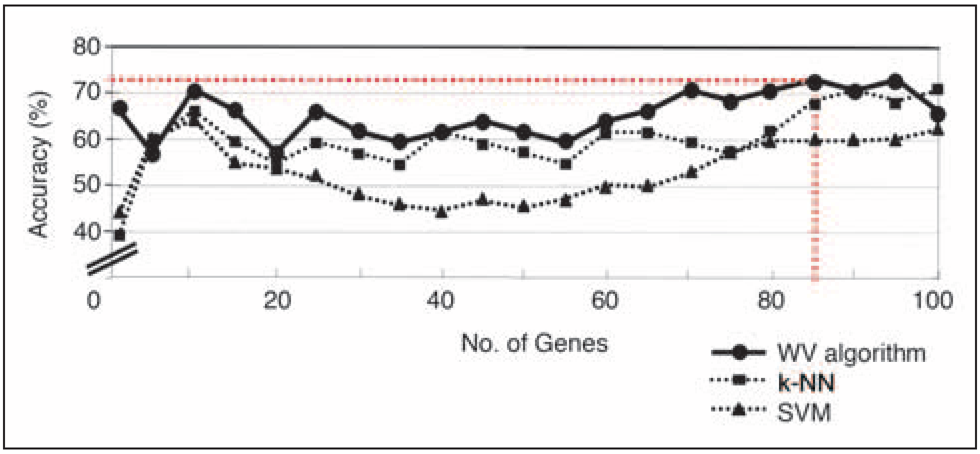


Fig. 2.1 Comparison of different algorithms in terms of classification accuracy (copied from Iwao-Koizumi et al., 2005)

However, as it is mentioned in their discussion part, very few genes (only 3) selected through their method is in common with the reported informative in previous studies. Although the authors stated that one main reason is that the criteria for discriminating two classes are different, their superior results should be verified.

2.4 Consistency gene selection method

2.4.1 The concept of consistency

As already mentioned in chapter 1, most gene selection methods in previous microarray studies were claimed very efficient, since the outcomes obtained from their experiments could achieve high accuracy. However, many of their experiments results can not be reproduced in practice, which has led them to be debated as spurious results. Recently, this issue has caught a lot of attention from researchers and is widely discussed in scientific journals (Ransohoff, 2005a, , 2005b). Consequently, in this thesis the concept of consistency is proposed, which is expected to solve this problem.

There is only one existing paper addressing the inconsistency problem in gene selection, since the concept is a new topic in microarray studies. Probabilistic consistency analysis for gene selection method (Mukherjee et al., 2004) is a recent novel approach that focuses on analyzing the common genes selected from two datasets. Consistency in this method is defined as follows:

Suppose two microarray datasets D_a and D_b targeting the same bioinformatics task, each having same number of genes. r is a ranking function generating two lists of sorted genes from the two datasets. Let s top-ranked genes in each case be selected and denoted by S_a and S_b. Then, the consistency C of this dataset is given by:

$$\mathrm{C}(r;\ s,\ D_a,\ D_b) = |\ S_a \cap S_b\ | \tag{2.23}$$

Consistency C is the number of genes in common between two datasets, and depends on ranking function, data and number of selected genes (Mukherjee et al., 2004). Hence, the greater the value of C, the higher the consistent of dataset.

Mukherjee et al. have applied this concept of probabilistic consistency to their gene selection method for selecting truly differentially expressed genes under various conditions. Their algorithm is called data-adaptive method (DA), as the gene selection function is optimized by the consistency based on selected genes in each run. DA gene selection algorithm is summarized in the two following stages:

1. Selecting genes in terms of consistency:
    a. D is resampled into a pair of D_a and D_b by using bootstrap algorithm, each having as many samples and genes as D.
    b. Testing statistical function F_t is applied to D_a and D_b for selecting two lists of top-ranked genes denoted by S_a and S_b respectively. f^* is defined as:

$$f^* = \arg\max_{f \in \mathscr{F}} C(F_t, S, D) \tag{2.24}$$

where $\mathscr{F}$ is a family of test function f and optimized by the data-adaptive algorithm that is described in stage 2.

c. The consistency *C* is calculated by Equation (2.23).

d. Repeat steps from a to c for *n* times (*n* is a pre-specified number, normally several hundreds).

Hence, a group of top-ranked genes are selected out based on the largest value of consistency *C* after *n* iterations.

2. Data adaptive optimized algorithm: using the selected genes to optimize the testing statistical function ***f***. Thus, the Equation (2.24) can be rewritten as:

$$f = \left\{ F_t \; \middle| F_t = \frac{\left|\overline{X_i} - \overline{Y_i}\right|}{\theta_1 \times \beta_i + \theta_2}, \theta_1 \in \{0,1\}, \theta_2 \in \{0,5\} \right\} \quad (2.25)$$

where:

$\overline{X_i}$: the mean value of ith gene corresponding to classes 0.

$\overline{Y_i}$: the mean value of i^{th} gene corresponding to classes 1.

β_i : the pooled standard deviation of the i^{th} gene.

θ_1, θ_2 : the constant parameters for Data-adaptive gene selection function.

Mukherjee's experiments have shown that the best consistency with the parameters (θ_1 and θ_2) occurs after 500 iterations (Mukherjee et al., 2005). Consequently, an optimized gene selection function can be constructed with the optimized parameters θ_1 and θ_2. Performed on their simulating dataset from Affymetrix arrays (Santa Clara, CA), DA gene selection method outperforms SAM and T-test statistic in their experiment.

The limitation of Mukherjee's work is that DA method was only performed on a simulating dataset. Theoretically, the algorithm of DA method is robust and can be used for gene selection on various microarray datasets. However, practically the method needs more benchmark datasets for evaluation. A simulating experiment of gene selection using T-test, SNR and DA algorithms is presented in chapter 4.

Chapter 3
Methodology

3. 1 Background

As discussed in Chapters 1 and 2, DA gene selection method (Mukherjee, Roberts, & Lann, 2005) is a recent novel approach that focuses on analyzing the common genes selected from two non-overlapped patient subsets of a microarray dataset. The concept of "consistency" proposed by Mukherjee and his colleagues is defined as the number of common genes selected from two sampled datasets. However, it is not clear to what extent the selected "highly differentially expressed genes" are positively correlated to the consistency of final classification performance, i.e. the performance of classification over individual sampling subsets of a complete dataset may still have a very inconsistent result, even though the common-gene method is employed.

In practice, the classification performance is a commonly accepted criterion for evaluating gene selection, while a valid performance is estimated in a k-fold cross validation policy. For example, a gene selection has a good performance over one 2-split of the whole dataset, where one subset is for training, the other subset is for testing. An important condition is that gene selection method is allowed only to run on the training set, and then applied the obtained genes to the classification of the testing set. A gene selection method can perform rather successfully on one 2-split trial, but for a method with low consistency, in many cases it may fail to achieve satisfactory results on another 2-split trial of the fold cross-validation. Therefore, except for the performance (i.e. classification accuracy of microarray diagnosis based on microarray classification), the consistency of gene selection in terms of performance is an important measurement for gene selection.

3.2 Motivation

With validation techniques (e.g. k-fold, leave-one-out cross validation), the result obtained over a training subset (self testing result) are often found very different from that over a testing subset (applying the trained model to the testing set). This means the consistency capacity of gene selection is fairly low, which results in the experimental results varying significantly. The ideal classification should be like this: provided with a k-fold cross validation technique, the accuracy of classification on one fold testing set is expected to be close to that on the remaining k-1 folds training set. These selected genes thus can be regarded as the informative or differentially expressed genes in terms of such high consistency and the achieved good classification accuracy. Otherwise, even if the accuracy of classification on the training set is 100%, the result on the testing set is not convincing, because the experiment is not repeatable on an independent testing dataset.

Therefore, for an optimal gene selection, whatever sampling of a microarray data, the performance of classification over the sampled datasets can reach a reasonably good performance consistently. In other words, for a microarray dataset, there exists a subset of genes such that the classification in the space spanned by this subset of genes enables a good classification performance on every possible train-and-test trial.

Consider a microarray dataset D having two predictor classes, class 1 and class 2 representing healthy and diseased patients, respectively. D_{sub} is a resampled subset from D. For every D_{sub}, if all samples in D_{sub} can be classified with the same accuracy, thus D can be seen as an ideal microarray dataset for classification task. In short, whatever D is resampled, the performance obtained from classification over D is consistently good. Fig. 3.1 shows an example of a 2-class distribution with the best consistency regarding to classification. Class 1 and class 2 are scattered in two distinct clusters with no class mixture and overlap between each other. Such type of data distribution is easy to handle by a simple classification modeling in that:

1) It can give the good classification accuracy for most typical classifiers, because the data with high correlations is separated into two clusters, which is easy for classifiers to distinguish.

2) Classification over one partition of data is consistent to that over the whole dataset.

However, very few real microarray datasets have such good consistency characteristics. The huge dimensionality of microarray data with many noise genes can confuse the classifiers, and the distribution of resampled data is normally scattered randomly. As a result, in most real microarray experiments, the classification performance varies significantly if the dataset is resampled several times. Motivated by this, a new gene selection method is aimed to developed, which is able to satisfy the above two criteria.

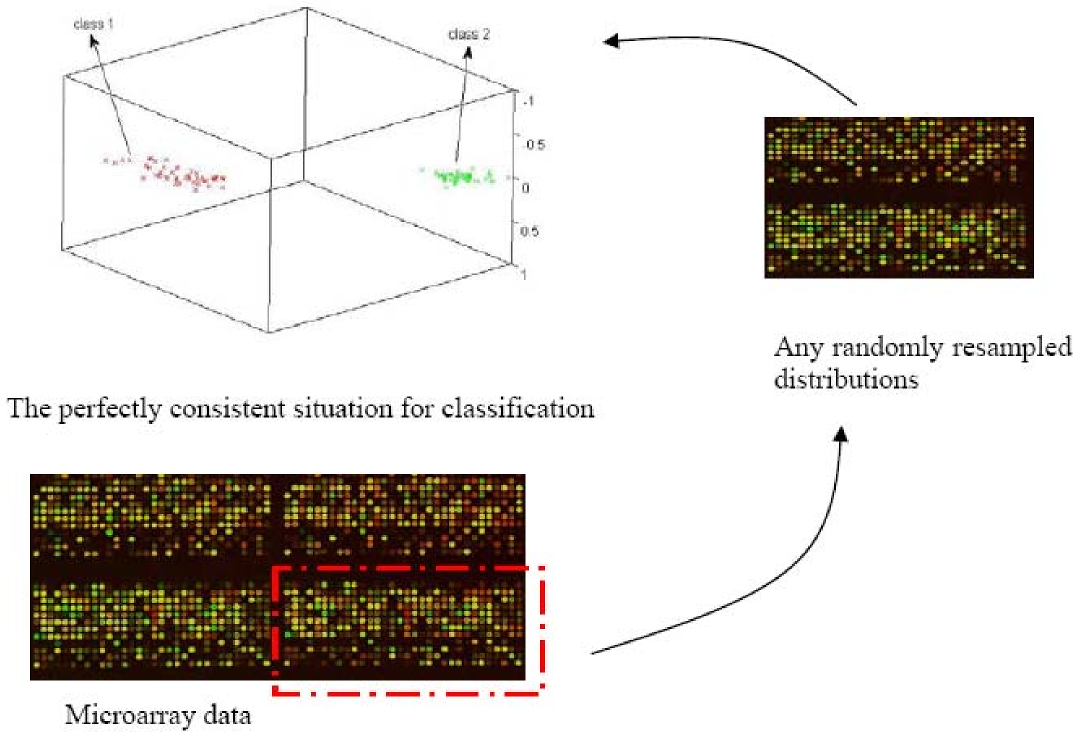

class 1
class 2
-1
0.5
0
0.5
1
Any randomly resampled distributions
The perfectly consistent situation for classification
Microarray data

Fig. 3.1 The microarray dataset with “perfect consistency”

3.3 Definition of Consistency

Consider a dataset pertaining to a bioinformatics task (two classes) and denoted by *D*. The dataset *D* consists of *n* samples with *m* genes, and all samples belong to two classes (e.g. class 1 or class 2). D_a and D_b are two subsets of *D* obtained by random subsampling, and serve as training and testing data, respectively.

$$D = D_a \bigcup D_b \quad \& \quad D_a \bigcap D_b = \varnothing \tag{3.1}$$

Given a base function $\boldsymbol{F}$ over *D*, and a gene selection function f_s over D_a, The consistency of dataset *D* can be calculated as

$$C(\boldsymbol{F}, f_s, D) = | P_a - P_b | \tag{3.2}$$

where P_a and P_b are the outcome of the function $\boldsymbol{F}$ on D_a and D_b,

$$P_i = \boldsymbol{F}(f_s(D_i), D_i) \mid i = a, b. \tag{3.3}$$

Base function $\boldsymbol{F}$ can be any of various data processing models, such as clustering function, partitioning function, feature extraction function, classification function, etc., it determines the feature space on which the consistency is based on. In the concept of consistency based on performance, $\boldsymbol{F}$ is set as one type of classification function.

3.4 Consistency in terms of classification performance

As $\boldsymbol{F}$ is assigned as a classification function, the above fundamental consistency definition Eq. (3.3) can be extended as,

$$P_a = \boldsymbol{F}(f_s(D), D_{a,} D_b)\text{ , and } P_b = \boldsymbol{F}(f_s(D), D_{b,} D_a) \qquad (3.4)$$

Substitute Eq. (3.4) into Eq. (3.2), Eq.(3.2) can be extended as a definition of consistency in terms of classification performance,

$$C(\boldsymbol{F}, f_s, D) = |\ \boldsymbol{F}(f_s(D), D_{a,} D_b) \quad - \boldsymbol{F}(f_s(D), D_{b,} D_a)\ | \qquad (3.5)$$

where $f_s(D)$ specifies D as the dataset for gene selection. D_a in the first term of Eq.(3.5) is assigned for classifier training, and D_b is for testing. The second term of Eq. (3.5) specifies a reversed training and testing position for D_a and D_b , respectively. Fig. 3.2 illustrates the procedure of computing Eq. (3.5). First, the performance P_a is computed by one classification on subset D_a. Then, P_b is obtained by another classification on subset D_b. Hence, a smaller C value represents a more consistency gene selection.

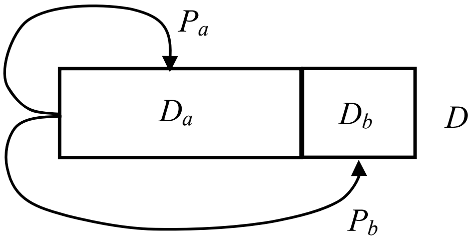


Fig. 3.2 Procedure of computing consistency (Form1)

Alternatively, Eq. (3.6) is another form of the performance-based consistency definition, which is obtained by replacing training and testing set in Eq. (3.5).

$$C(\boldsymbol{F}, f_s, D) = |\ \boldsymbol{F}(f_s(D), D_{a,} D_a) \quad - \boldsymbol{F}(f_s(D), D_{a,} D_b)\ | \qquad (3.6)$$

Fig. 3.3 shows the procedure of computing Eq. (3.6). Here, the classifier is trained on D_b, and then the performance is computed by the classifier on the other subset D_a.

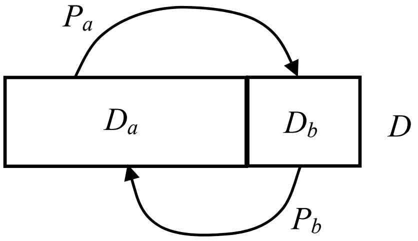


Fig. 3.3 Procedure of computing consistency (Form2)

3.5 The proposed GAGSc gene selection method

In this study, a new method called GAGSc is proposed for gene selection, in which two different algorithms are used for searching and evaluating the informative genes, such as Genetic Algorithm (GA) and K-Nearest Neighbor (KNN). GAGSc method is an optimizing computation taking classification consistency as an additional measurement for gene selection.

3.5.1 Introduction of GA

GA has been widely applied in the areas of science and engineering as the learning-adaptive algorithms for solving complex optimization problems. Unlike traditional statistical algorithms, GA does not require extensive knowledge of the learning area. Generally, with sufficient generations, GA can converge towards an optimized outcome that is often superior to the performance of traditional statistical algorithms. The principle of GA is based on the simulation of the process in natural evolution, following the procedure of survival of the fittest proposed by Charles Darwin. In GA, an intelligent search engine randomly explores a defined space to find an optimized solution.

Generally GA has five components, namely chromosome, fitness function, selection, operation (mutation, crossover, etc.) and stopping criterion. In the context of gene selection, an initial set of individuals (genes) are randomly generated from the whole population and is called 'chromosome' within the process of GA. The fitness of individuals in this chromosome is measured by a fitness function, and a subset of genes from the chromosome is selected out based on their fitness scores. Then, the GA operations (such as mutation and crossover) give a new generation that replaces the parent chromosome. The process will be iterated until the stopping criterion is reached. In practice, certain sophisticated evaluation criteria can take into account the combination of individual genes. Here, GA algorithm is simply summarized into the following steps:

1. Randomly select an initial set of genes to create a chromosome.
2. Compute the usefulness of each individual through an evaluating function (in GA, it is known as "fitness function").
3. Select the candidates for new generations by using genetic operators, such as mutation and crossover.
4. Create a new generation consisting of the candidates that satisfy the selection

criteria.

5. Repeat from step 2 until a satisfactory solution is reached.

The process usually iterates hundreds of times, each of which is called a generation. The final results will often highly fit the selection criteria (Mitchell, & Forrest, 1994).

Mutation and crossover are two well-known inductive operators in GA for individual selection. In real GA applications, there are many approaches to perform mutation and crossover. In this chapter, a simple example is given as follows for briefly describing these operations:

Given a sequence (from 1 to 9), the mutation operator changes the order several items in this sequence to create a new sequence:

(1 2 ***3*** 4 5 6 8 ***9*** 7) => (1 2 ***9*** 4 5 6 8 ***3*** 7)

where the sequence (1 2 ***9*** 4 5 6 8 ***3*** 7) is a child generation created by mutation.

In crossover operation, two parents sequence and a crossover point are required. The child sequence is copied from the first parent till the crossover point. Then, the items in other parent will be added into the child sequence, if they are not in the offspring.

(**1 2 3 4** ***5*** 6 7 8 9) + (5 4 3 6 9 8 7 1 2) = (**1 2 3 4** ***5*** 6 9 8 7)

where ***5*** is a crossover point. Note that there are different ways to present crossover operation in real GA applications.

3.5.2 Related work of GA for gene selection

Since GA has been regarded as an effective approach for searching complex multi-dimensional space (Goldberg, 1989; Holland, 1975), it has been applied to different pattern recognition problems, including gene selection. Huerta et al. (2006) proposed a GA/SVM gene selection method that reportedly achieved very high classification accuracy (99.41%) on colon data (Alon, Barkai, Notterman, Gish et al., 1999). However, their good result was not obtained from an independent validating dataset, but on the same dataset that was used for classifier training, which means the reproducibility of their experiments should be questioned. Fig. 3.4 shows the process of the GA/SVM gene selection method (Li, Weinberg, Darden, & Pedersen, 2001). Li et al (2001) introduced a GA/KNN gene selection method for sample classification. Their method was performed on colon and leukemia datasets, and the experiment results were based on k-fold cross validation. They claimed that the GA/KNN method was capable of finding a set of informative genes from the original data, and the selected genes were highly repeatable. The main idea of their method is to use a classifier to estimate the importance of subsets of genes, and then select those genes that most frequently appear in the nearly optimized subsets. The limitation of this study is that as the bias error estimation and the classification result on the independent validating test are not mentioned in their paper, it is hard to evaluate the effectiveness of the GA/KNN method. The flowchart of GA/KNN method for gene selection is illustrated in Fig. 3.5.

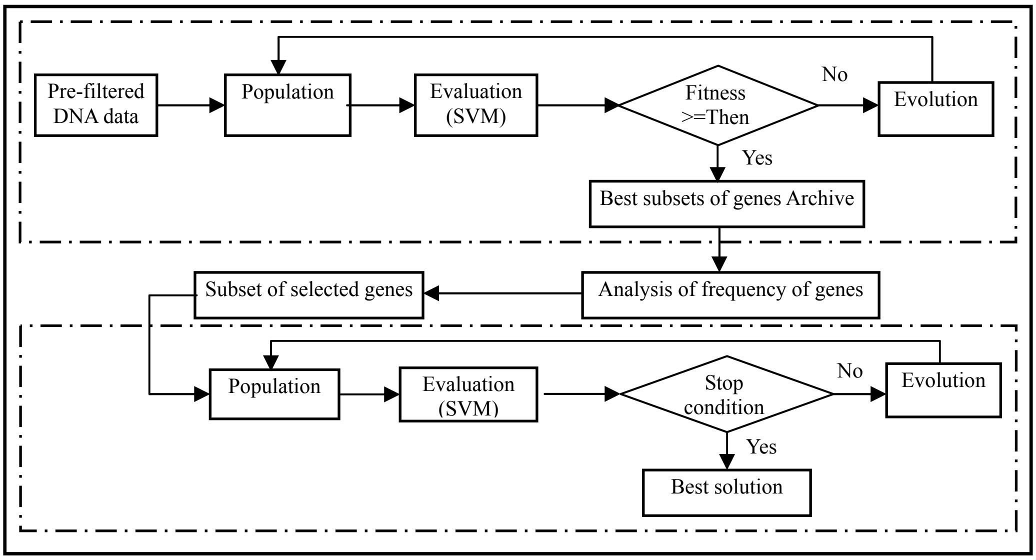


Fig. 3.4 A brief flowchart of gene selection process in GA/SVM method (adapted from Huerta et al. 2006)

GA is a robust approach for those difficult multi-dimensional problems, and used in a wide variety of optimization tasks (Hopfield, & Tank, 1985; Louis, 1993). It is particularly effective and suggested to use when the research task has the following characteristics:

1) The search space is huge, complex or lacking sufficient information.
2) Poorly-understood knowledge in the target research domain.
3) Traditional search methods are failed to perform the task.

The main drawbacks of GA are in the difficulties of developing a fitness function and determining a stopping criterion. The performance of GA application is heavily dependent on the suitability of the fitness function. The final result cannot be guaranteed towards a global optimization without an effective fitness function. As GA usually involves a revolutionary process, whether the result can achieve a satisfactory level is highly correlated with a good stopping criterion. However, these two above issues are not easy to handle, because they are sensitive to the conditions of the target bioinformatics problems.

GA/KNN gene selection:

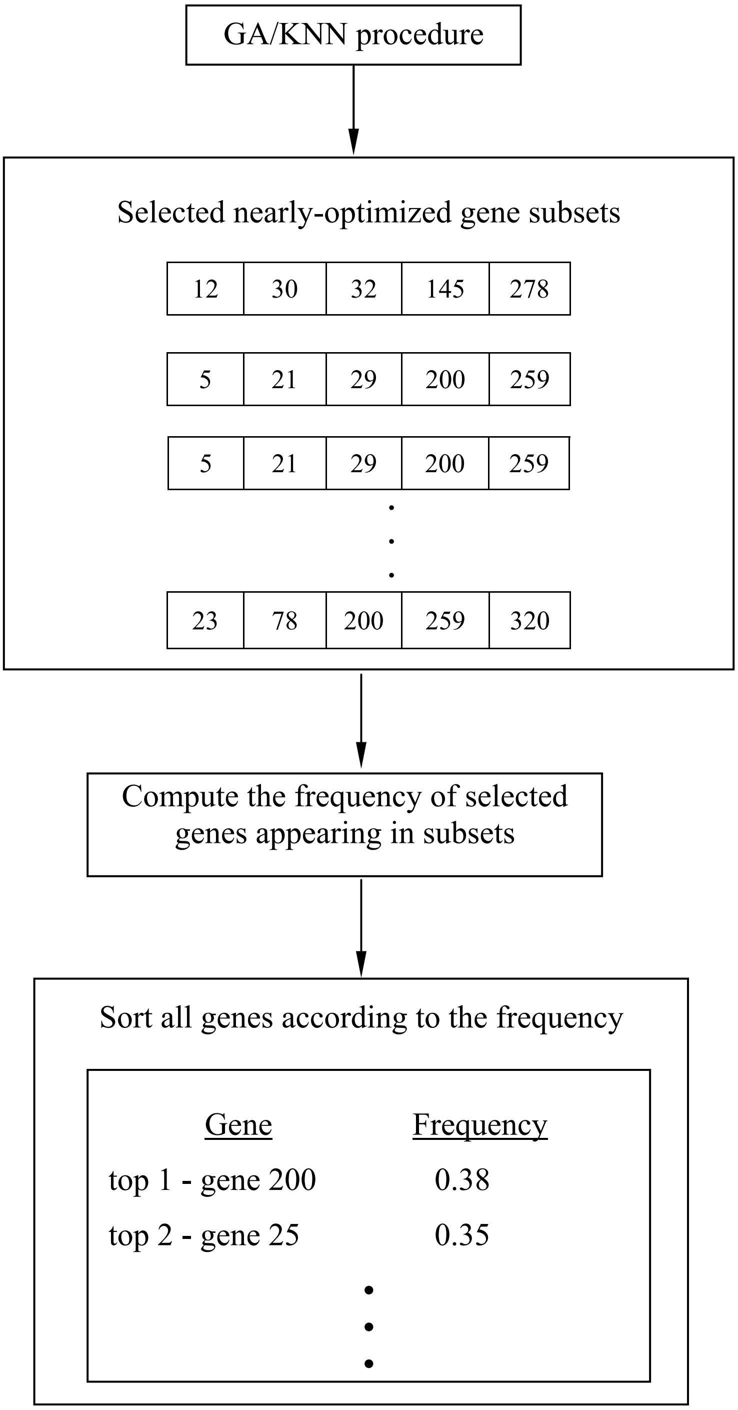

GA/KNN procedure
Selected nearly-optimized gene subsets
12 30 32 145 278
5 21 29 200 259
5 21 29 200 259
23 78 200 259 320
Compute the frequency of selected genes appearing in subsets
Sort all genes according to the frequency
Gene
Frequency
top 1 - gene 200
0.38
top 2 - gene 25
0.35

Fig 3.5 A flowchart of GA/KNN gene selection (Adapted from Li, et al., 2001). Note that the number of candidate subsets is supposed as 5.

3.5.3 The proposed GAGSc algorithm

The key idea of GAGSc algorithm proposed in this work is using the result of consistency in terms of performance obtained from an operation (e.g. classification or clustering) on a microarray dataset. So far, there is no agreement on which genes are highly differentially expressed, and consequently it is difficult to measure the reliability of any gene selection method. In practice, the performance of an operation over microarray data is a straightforward criterion for measuring the outcomes of microarray experiments. The new solution is based on an optimizing computation that takes consistency into account.

In the proposed GA method, an evolutionary function is employed for selecting candidate genes. Mutation and crossover operators are applied to this evolutionary function for optimizing gene selection function. Mutation and crossover are adaptive heuristic search algorithms based on an evolutionary idea. The main strength of these operators is that they help solution converge towards the global optimum over sufficient successive generations. Meanwhile, they also can provide a fast, effective and robust search. The results obtained via GA often consist of new combinations of genes that contain more important implicit information than using simple top-ranked individuals. This new solution for gene selection does not require any prior knowledge about the microarray dataset. Given a dataset *D*, a list of genes *S*, and an operation function F_{sc}, the optimized function performing GA method is expected to achieve for selecting gene:

$$f_s^* = \arg\min_{f_s \in \mathscr{F}} C(F_{sc}, S, D) \tag{3.7}$$

where $\mathscr{F}$ refers to a family of evolutionary gene selection functions, F_{sc} and f_s refer to the function of computing consistency under the condition of gene selection and gene selection function, respectively.

Now, the algorithm can be simply summarized into the following steps:

1. Split all genes of dataset *D* into ρ segments based on their mean value.
2. Randomly select one gene from each of ρ segments, respectively. The initial candidate gene set contains ρ genes and is denoted by *S*.
3. Apply the operation function F_{sc} (e.g. classification) to the data containing those genes listed in *S*, and compute the consistency *C* by Eq. (3.5) or Eq. (3.6).
4. Perform gene selection function f_s on *S* to get a new generation of genes S', and compute the consistency C'.

5. If $C' < C$, then $C = C'$ and $S = S'$.
6. Repeat Steps 3-5 for N generations. N is a given number for determining how many generations are used in this case.
7. Output the finally selected genes.

The optimized gene selection method is obtained after N generations based on the best consistency performance. In each generation, D_a and D_b are resampled B times depending on the size of samples, for example, if the sample size of dataset is larger then 30, B is set to 50, otherwise 30. Consequently, C' is the mean value of the consistency scores for B rounds computation.

In practice, the evaluation of consistency is a multiple-objective optimizing problem, because there is a possibility that the improvement of consistency might be coupled with the deterioration of performance. This means that even if the consistency C of new generation of genes is better than its ancestor, the performance of classification P on microarray data might be worse. Therefore, in practice, a ratio to consistency and performance is accepted to balance them in the purpose of optimizing these two variables simultaneously. The ratio R is defined by:

$$R = \frac{C}{w \times P} \tag{3.15}$$

where w is a pre-defined weight for adjusting the ratio in experiment, and P is the classification performance on dataset D, and can be computed by Eq.(3.3).

In this sense, Eq.(3.7) can be rewritten as:

$$f_s^* = \arg\max_{f_s \in \mathcal{F}} R(F_{sc}, S, D) \tag{3.16}$$

For simplicity, a basic flowchart of GAGSc method is given in Fig. 3.6.

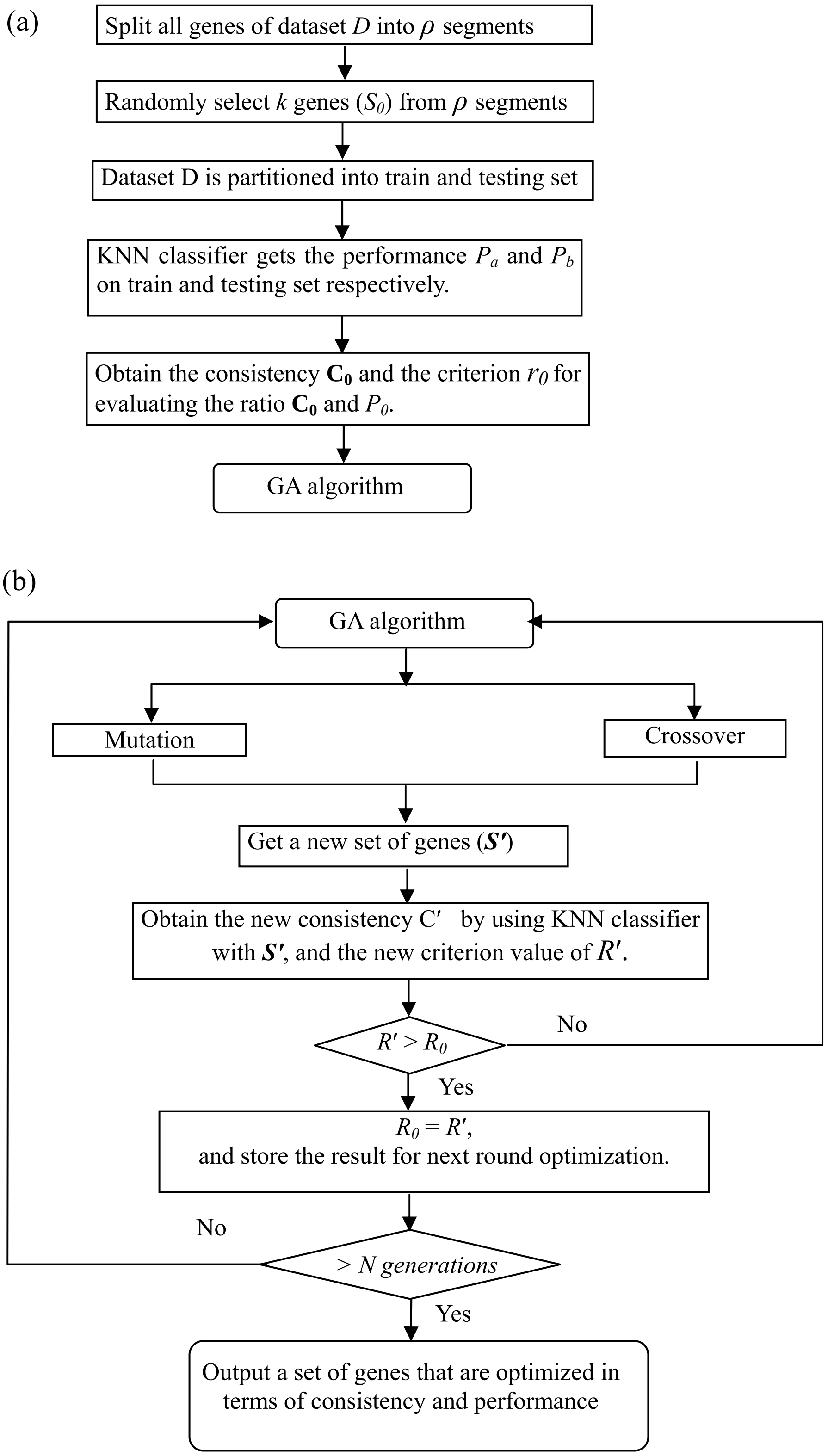

(a)
Split all genes of dataset D into ρ segments
Randomly select k genes (S_0) from ρ segments
Dataset D is partitioned into train and testing set
KNN classifier gets the performance P_a and P_b on train and testing set respectively.
Obtain the consistency $\mathbf{C_0}$ and the criterion r_0 for evaluating the ratio $\mathbf{C_0}$ and P_0.
GA algorithm
(b)
GA algorithm
Mutation
Crossover
Get a new set of genes ($\boldsymbol{S'}$)
Obtain the new consistency C′ by using KNN classifier with $\boldsymbol{S'}$, and the new criterion value of R'.
$R' > R_0$
No
Yes
$R_0 = R'$, and store the result for next round optimization.
No
$> N$ generations
Yes
Output a set of genes that are optimized in terms of consistency and performance

Fig. 3.6 The flow chart of GAGSc method:
(a) The algorithm for initial gene selection.
(b) Evolution function for gene selection in GAGSc.

For clarity, a schematic example of the above GAGSc method with ratio *R* is also given as follows:

GAGSc gene selection algorithm (Pseudo code)

Function (1): Initial gene selection

```
    /* Create initial generation of genes: $S$ */
    Separate all genes into $\rho$ segments based on their mean value ;
    for   $i$ = 1 to   $\rho$
            Randomly select one gene $g$ from segment $i$ ;
    end
    $S \leftarrow$   $\rho$ genes ;
```

Function (2): KNN consistency computation

```
    for   $j$ = 1 to $B$            /* $B$ is the predefined times of resampling */
            Partition data $D$ into $D_a$, $D_b$ ;
            Calculate accuracy of classification $P_a$, $P_b$ with $S$ ;
            Compute consistency $C$ ;
    end
    Calculate mean of consistency $C$, and compute classification accuracy $P$ on $D$ ;
    Compute ratio $R$ ;
```

Function (3): Evolutionary gene selection:

```
    for   $i$ = 1 to $N$          /* $N$ refers how many generations the algorithm creates */
            Create new generation of candidate genes ($S'$) by mutation or crossover ;
            Compute consistency $C'$ by using $S'$ ;
            /* C' is the mean value of consistency vector computed by Function 2 */
            Compute ratio $R'$ ;
            If   $R' > R$
                  $S \leftarrow S'$;     $R \leftarrow R'$ ;
            end
    end
Output $S$       /* the selected informative genes */
```

3.6 The proposed LOOLSc gene selection method

LOOLSc gene selection method is derived from the Leave-one-out SVM (LOOSVM) algorithm that has been successfully implemented in gene selection for microarray data (Tang, Suganthan, & Yao, 2006; Zhao, & Kwoh, 2006; Zhou, & Mao, 2005). With this solution, the informative gene set starts from an empty set and a sequential forward selection (SFS) search engine adds genes iteratively (see section 3.7.3). The candidate genes are evaluated in terms of their performance computed by K-Nearest Neighbour (KNN) or support-vector-machine (SVM) classifier on the whole dataset.

In order to evaluate the effectiveness of using consistency concept in the proposed LOOLSc method, the published LOOLS method (without consistency measurement) is applied to the experiment for comparison. In LOOLS, a variation of SVM algorithm, Least-square SVM (LS-SVM) is adopted for evaluating genes without consistency criterion. In LOOLSc method, consistency is computed by LOOSVM algorithm, and candidate genes are measured by LOOE (Leave-one-out Error) algorithm.

3.6.1 Introduction of LOOE algorithm

In the proposed LOOLSc gene selection method, a criterion called LOOE is employed to measure candidate genes. LOOE is computed by LS-SVM classifier that is an enhanced version of standard SVM algorithm (Suykens, & Vandewalle, 1999). Without consistency criterion, the simple SVM evaluation criterion for gene selection might have generalization error and cannot be performed efficiently, *i.e.* although all samples can be correctly classified in training stage, they might be misclassified during the leave-one-out test stage if they are close to optimal hyperplane (see SVM classification algorithm section). Motivated by this issue, the LS-SVM classifier with a LOOE criterion is applied to LOOLSc gene selection algorithm.

The LOOE criterion based on LS-SVM (Tang et al., 2006; Zhou et al., 2005) can be formulated as follows:

Consider again Equation (3.14):

$$f(x) = w^T \cdot \mathrm{x} + b$$

After one sample x_i is removed in the LS-SVM training stage, the result of a test sample x in the Leave-one-out validation stage is denoted as:

$$y_i f^i(\mathrm{x}) = 1 - \frac{\alpha_i}{(\mathrm{H}^{-1})_i} \tag{3.17}$$

where y_i is the class label of the sample x_i, α_i is a scalar in the Lagrange multiplier vector, and H^{-1} is defined by:

$$H = \begin{bmatrix} K+\gamma^{-1}I & \vec{1} \\ \vec{1}^{T} & 0 \end{bmatrix} \tag{3.18}$$

where:

K is a kernel matrix;

γ is a given positive constant that is for adjusting generalization and training errors;

I is an identify matrix.

$\vec{1}^{T} = [1,1,...,1]^{T}$;

$(H^{-1})_i$ is the ith diagonal element of the matrix H^{-1};

Thus, the LS-B criterion of measure genes can be defined as:

$$\text{LOOE} = \frac{n - \sum_{i}^{n} \text{sign}\left(1 - \frac{\alpha_i}{(H^{-1})_i}\right)}{2n} \tag{3.19}$$

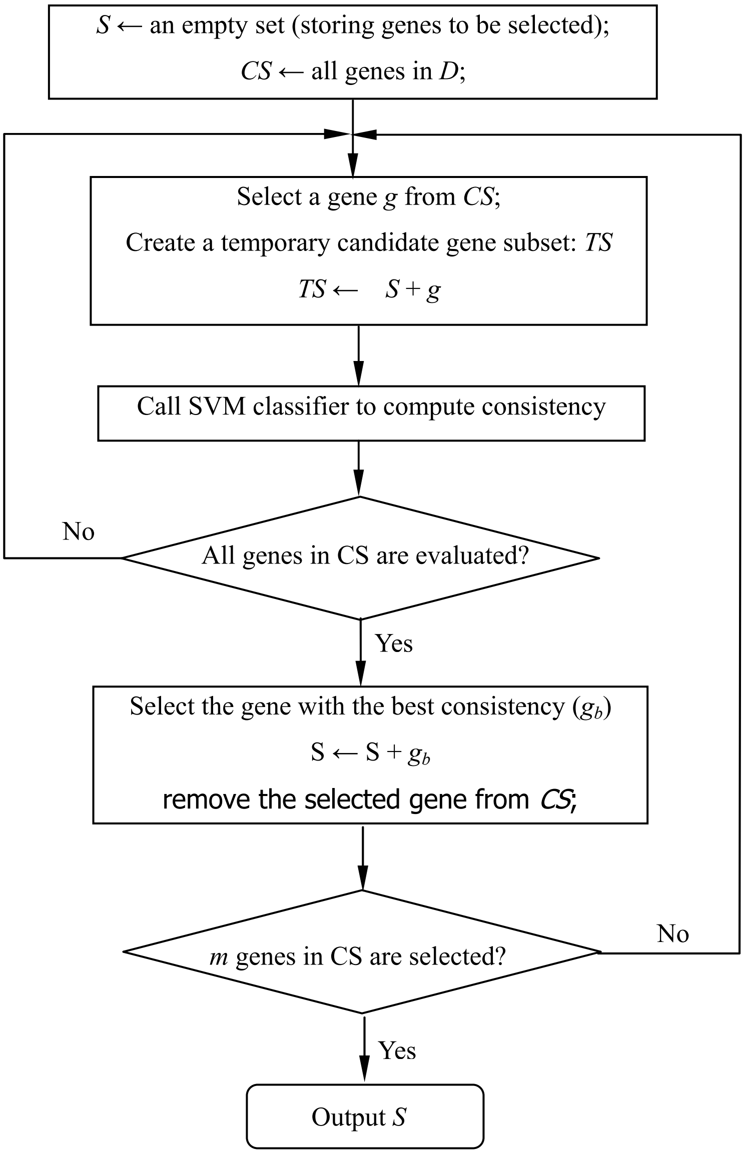
In short, with LOOE criterion, if the test sample x is misclassified, $y_i f^i$ (x)returns a negative value (-1), otherwise positive value (1). Therefore, in terms of LOOE criterion, the small absolute value of LOOE is preferable, which means the sample x is close to the optimal hyperplane and can be correctly classified.

3.6.2 Proposed LOOLSc algorithm

So far, the LOOE criterion derived form LS-SVM algorithm has been obtained and can be used in the proposed LOOLSc method. The LOOLSc algorithm for gene selection can be simply summarized as:

Suppose a dataset *D* with *n* genes, LOOLSc algorithm starts with the initialization of two gene sets, one is an empty set of aggregating informative genes and denotes it as *S*, the other is a candidate gene set with all genes from *D* and is denoted by *CS*. Next, SFS algorithm searches the space of candidate set *CS*, and sequentially selects one gene g_i and puts it and the genes in *S* into a temporary set *TS*. For each gene, the consistency is computed by a SVM classifier using the genes of *TS*, so that a matrix with *n* consistency value can be generated after one round. Then, the gene with the best consistency is selected from *CS* and put into set *S*. At the same time, this gene is removed from the candidate set *CS*. After the first round is finished, there should be one gene in *S* and it is granted as an informative one in terms of consistency.

Now, SFS sequentially selects the next gene from set *D* and puts it and the existing genes in *S* into *TS* to do classification through SVM. One gene with best consistency can be found and becomes the next informative gene is set *S*. The second gene is then removed from *CS*. The selection procedure will repeated until *m* (a pre-defined number) genes are selected out as the informative genes. For clarity, a flowchart of LOOLSc method is presented in Fig. 3.7.



$S \leftarrow$ an empty set (storing genes to be selected);
$CS \leftarrow$ all genes in D;
Select a gene g from CS;
Create a temporary candidate gene subset: TS
$TS \leftarrow S + g$
Call SVM classifier to compute consistency
No
All genes in CS are evaluated?
Yes
Select the gene with the best consistency (g_b)
$S \leftarrow S + g_b$
remove the selected gene from CS;
m genes in CS are selected?
No
Yes
Output S

Fig. 3.7 The flowchart of LOOLSc gene selection method, where, *m* is a pre-defined number of genes to be selected.

For clarity, the LOOLSc algorithm for gene selection is modelled as following pseudo code:

Function (1): Initial gene sets:

```
    /* Initialization */
    $S \leftarrow$ an empty set ;      /* $S$ is for storing the informative genes to be selected */
    $CS \leftarrow$ all genes in $D$;
```

```
Function (2): Select candidate genes
    for  $i$ = 1 to $m$            /* $m$  is a pre-defined number of genes to be selected */
        for j = 1 to $q$      /* $q$ is the number of genes in $CS$ */
            Sequentially select one gene $g$  from $CS$;
            Create a temporary gene set $TS$;
            $TS \leftarrow S + g$;
            Call SVM consistency computation function      /* function (3) */
        end

        $S \leftarrow$ the gene with the best consistency ($g_b$) + $S$;
        remove the selected gene from $CS$;
    end
    output $S$ /* $S$ contains the final selected informative genes */
```

```
Function (3):  SVM consistency computation
    for  $j$ = 1 to $B$           l* $B$ is the predefined times of resampling */
        Partition data $D$ into $D_a$, $D_b$;
        Using SVM classifier to calculate performance $P_a$, $P_b$ with all genes in $TS$;
        Compute consistency $C$;
    end
    Calculate the mean of consistency $C$.
```

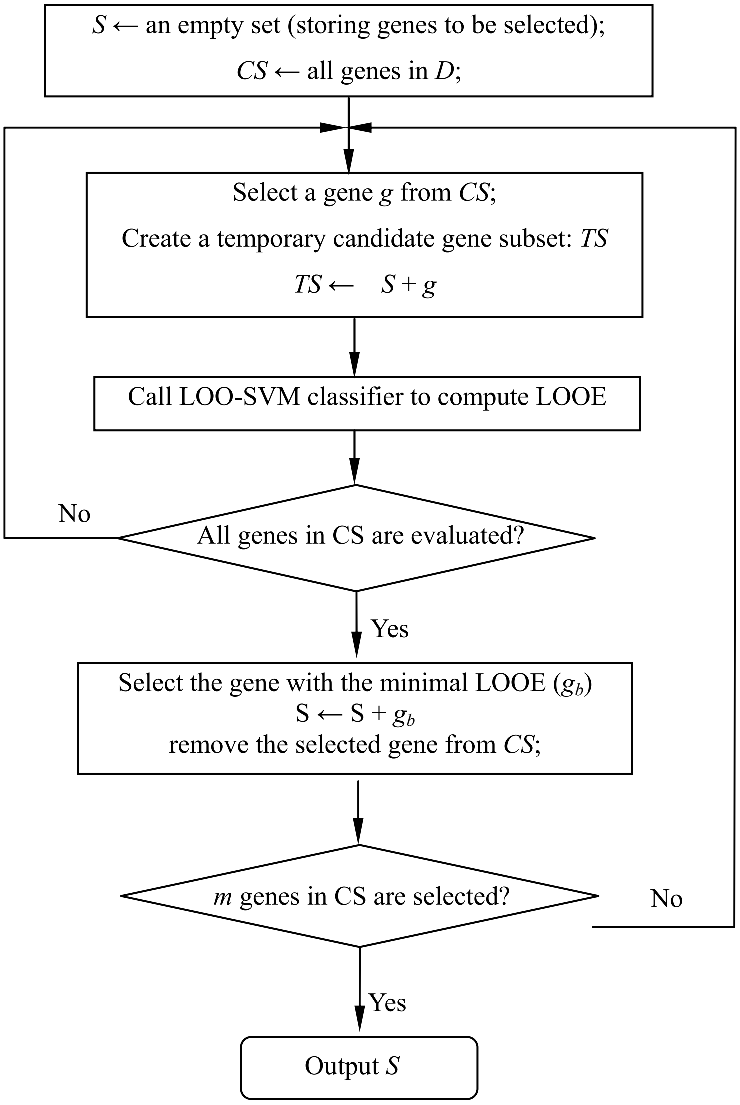
3.6.3 LOOLS gene selection algorithm

In order to evaluate the validity of using consistency concept in LOOLSc gene selection method, a gene selection method (LOOLS) proposed by Tang et al. (2006) is adopted for comparison. This method is based on the same criterion LOOE for measuring the importance of genes, but without using consistency concept. The LOOLS gene selection algorithm is shown in a flow chart in Fig. 3.8.

For clarity, a schematic model of LOOLS gene selection method is given as follows:

```
Function (1): Initial gene sets:
    /* Initialise selected informative gene set */
    $S \leftarrow$ an empty set ;
    $CS \leftarrow$ all genes in $D$;
```

```
Function (2): Select candidate genes
    for  $i$ = 1 to $m$            /* $m$  is a pre-defined number of genes to be selected */
    for j = 1 to $q$      /* $q$ is the number of genes in $CS$ */
      Sequentially select one gene $g$ from $CS$;
      Create a temporary gene set $TS$;
      $TS \leftarrow S + g$;
      Call LOO-SVM classifier to compute LOOE      /* function (3) */
    end
    $S \leftarrow S$ + the gene with the minimal LOOE ($g_b$) ;
    remove the selected gene from $CS$;
end
output $S$;      /* $S$ contains the final selected informative genes */
```



$S \leftarrow$ an empty set (storing genes to be selected);
$CS \leftarrow$ all genes in D;
Select a gene g from CS;
Create a temporary candidate gene subset: TS
$TS \leftarrow S + g$
Call LOO-SVM classifier to compute LOOE
No
All genes in CS are evaluated?
Yes
Select the gene with the minimal LOOE (g_b)
$S \leftarrow S + g_b$
remove the selected gene from CS;
m genes in CS are selected?
No
Yes
Output S

Fig. 3.8 The flowchart of LOOLS gene selection method. Note, *m* is a pre-defined number of genes to be selected.

3.7 Base functions and relevant algorithms

In the proposed GAGSc and LOOLSc, K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) are two base functions used to compute the consistency in terms of performance. KNN and SVM are two popular algorithms for solving pattern recognition problems in the domain of machine learning. Recently, they have been successfully used in different methods for gene selection.

In the present work, only KNN and SVM are used in the computation of consistency. However, according to the fundamental consistency definition Eq. (3.3), consistency can be computed through any other classification function, such as C4.5 trees, MLP, and traditional statistical classification methods, etc.

3.7.1 K-Nearest Neighbor (KNN)

KNN, since first introduced by Fix and Hodges (1951), has been extensively studied and discussed in respect to classification and clustering in the areas of pattern recognition, including microarray data analysis. Different types of KNN and its extensions have been proposed, Weinberger et al. (2005) presented large margin nearest neighbor (LMNN) for distance metric learning, Cui et al. (2003) introduced a tree-based KNN search algorithm for high-dimensional data analysis, and Xia et al. (2005) proposed a Reverse K-Nearest neighbors (RKNN) algorithm in profile-based system. In recent years, KNN, one of the simplest and best-known classifiers has been adopted in bioinformatics research, including microarray data analysis, such as Crimins et al. (2002) used KNN algorithm to do classification on lung cancer data, Binder et al. (2005) applied KNN algorithm to a novel method for their immunoassay-based anti-nuclear antibody test, and Kim et al. (2004) applied a sequential K-nearest neighbor (SKNN) method for dealing to their microarray experiments. All these experiments have shown KNN algorithm can yield competitive results in different applications.

The principle of KNN classifier is that similar objectives belong to similar class. Most KNN classifiers use standard *Euclidean distance* to measure the similarity between objectives:

$$d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \tag{3.7}$$

where x, y are the objectives.

In KNN classification, the given dataset is first split into the training set and testing set that are denoted by $x = \{ x_1, x_2, \dots x_n \}$, and $y = \{ y_1, y_2, \dots y_n \}$, respectively. The samples of the training and testing sets are identified by the class labels $C = \{C_1, C_2\}$ (Note that the samples in most microarray datasets belong to two classes). Suppose y_i is a sample whose class label is unknown. The K nearest neighbor samples in x according the distance calculated by Eq. (3.7) can be found and denoted by $Nset_{(k)} \in x$. Then the most common class label C_i appearing in $Nset_{(k)}$ is assigned to y_i. Fig. 3.9 shows a simple sample for computing K nearest neighbors.

In the case that more than one class label occurs with equal frequency in k nearest neighbor set Nset(k), the KNN test run on K-1 classifiers. This is a recursive process until there is only one class represented in the result.

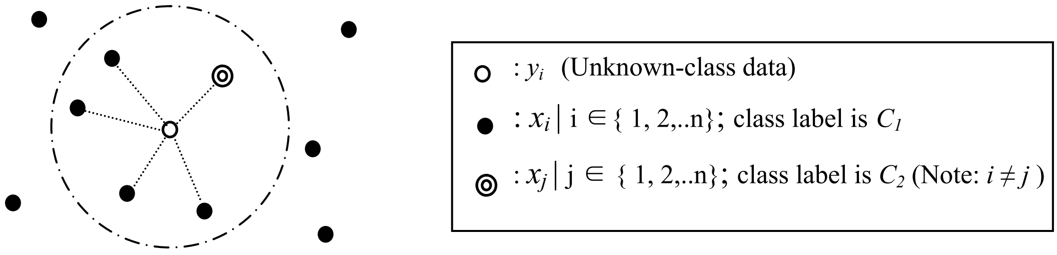

○ : y_i (Unknown-class data)
● : x_i | i ∈ { 1, 2,..n}; class label is C_1
◎ : x_j | j ∈ { 1, 2,..n}; class label is C_2 (Note: $i \neq j$)

Fig. 3.9 A simple schematic sample for computing K nearest neighbors. In this case, K = 5. The unknown-class data y_i is classified into class 1 (C_1), because 4 out 5 nearest neighbors belong to class 1.

KNN algorithm in GAGSc is used to do classification on microarray data for computing consistency in terms of performance. In KNN classification, K nearest neighbor genes in the training set responding to each gene in the testing set can be found and aggregated. According to the criterion of choosing class, each gene in the testing set is assigned to different classes. Then, the performance of classification can be reached and used for final consistency computation.

Although KNN classifier has been widely used in pattern recognition, it does have certain disadvantages that are criticized by researchers. For example: (a) it takes long computation time to train data; (b) it is difficult to choose the optimized K value; (c) it can be confused by irrelevant data.

3.7.2 Support Vector Machine (SVM)

SVM, introduced by Vapnik (1998), has been widely used as an efficient tool for classification and regression problems in pattern recognition research. Different types of SVM algorithms have been presented, such as Least Square Support Vector Machine (LSSVM) (Suykens et al., 1999), implicit Lagrangian Square Support Vector Machine (LSVM) (Mangasarian, & Musicant, 2001), Newton Support Vector Machine (NSVM) (Fung, & Mangasarian, 2004) and SVM Classification Tree (Pang, Kim, & Bang, 2005). The key idea of SVM algorithm is using linear classification techniques to solve non-linear classification problems.

SVM classifiers are generally binary-based. If the data is linearly distributed, SVM computes the hyperplane that maximizes the margin between the training samples and the class boundary. In contrast, if the data is not linearly distributed, the samples are mapped to a multi-dimensional space in which such a separating hyperplane can be constructed. A simple example of linear separate hyperplane of SVM classifiers is given in Fig. 3.10. This mapping process is usually called the kernel function (Huerta, Duval, & Hao, 2006). Such mechanism of SVM makes it a powerful classifier with good performance in microarray data analysis for reducing redundant genes (Guyon, Weston, Barnhill, & Vapnik, 2002; Mukherjee, 2003).

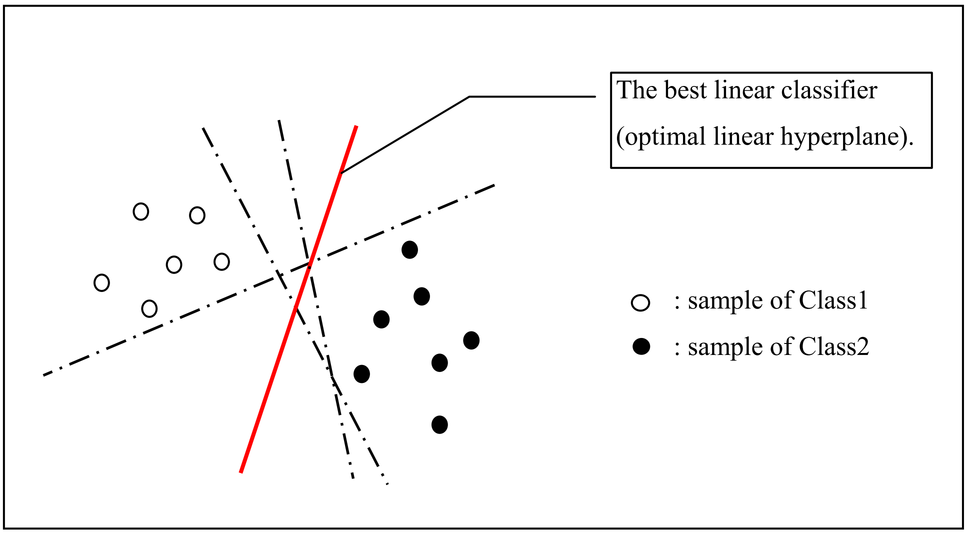


Fig 3.10 A simple example of linear separate hyperplane of SVM classifiers (Adapted from Gunn , 1997). In this space, there are many potential linear classifiers that can successfully separate points (samples), but only one can maximize the margin (distance) between two nearest points belonging to different classes. The best linear classifier is commonly called optimal linear hyperplane.

Mathematically, a typical SVM can be formulated as the following model (Gunn, 1997): Consider a dataset D pertaining to a 2-classes classification task and modelled as:

$$D((x_1,\mathrm{I}_1),\ (x_2,\mathrm{I}_2),\dots,\ (x_n,\mathrm{I}_n))\ |\ x \in D,\ \mathrm{I}\ \in\ \{-1,\ 1\} \tag{3.8}$$

where x_i is a vector (contains a number of genes in microarray data analysis), and I is a class indicator that denotes the class label of x_i. In this case, the given dataset D consists of samples x_i with associated "real" class labels $\mathbf{I}_\mathrm{i}$ can be seen as a training set. The linear hyperplane rule of SVM is defined as:

$$w^T \cdot \mathrm{x} + b = 0 \tag{3.9}$$

where $w = [w_1, w_2, \dots, w_n]^\mathrm{T}$, and b is a scalar. Both of them are constrained by the optimizing function as follows:

$$\mathrm{W}(\alpha) = \min \boldsymbol{L}(w, b, \alpha) \tag{3.10}$$

where $\boldsymbol{L}$ is a Lagrange function, and α is a Lagrange multiplier.

Since in the optimal hyperplane that separates those closest vectors belonging to different classes, linear hyperplane in Eq. 3.9 can be re-written as

$$w^T \cdot \mathrm{x} + b = 1 \tag{3.11}$$

$$w^T \cdot \mathrm{x} + b = -1 \tag{3.12}$$

The vectors are optimally split by the hyperplane (in Eq. 3.11, Eq. 3.12), if the distance between closest vectors belonging to two different classes is maximal. With Eq. 3.10, the w, α and optimal hyperplane is given by:

$$w = \sum_{i=1}^{n} \alpha_i x_i y_i \tag{3.13}$$

The classifier is thus defined as:

$$f(x) = w^T \cdot \mathrm{x} + b \tag{3.14}$$

The result of f(x) (either 1 or -1) is assigned to the test sample x as the class label.

Comparing with traditional algorithms, such as statistical methods and neural networks, SVM classifier can usually converge to a global optimization. Another main advantage is it usually outperforms other simple classification algorithms, for example, KNN classifiers. Hence, SVM algorithms are suggested to be used in the area of high dimensional data analysis, such as microarray data (Furey et al. 2000). However, it is not a perfect algorithm and inevitably has certain issues. The main limitations include the difficulty in choosing kernel functions (Burgess, 1998), and fairly slow computation speed both in training and test data.

Leave-One-Out SVM classifier is used in LOOE method to calculate the performance of classification on a subset with candidate genes. A polynomial kernel in SVM classifier is chosen in this work, and a Leave-One-Out (LOOCV) SVM method is applied to calculate the average accuracy on training data. The test data only have one sample and the SVM classifier is trained on all other samples.

3.7.3 Searching mechanism in gene selection

A basic component in gene selection is the searching mechanism that generates candidate gene subsets for evaluation. Generally, there are three types of searching schemes, random searching, sequential searching and evolutionary searching. In the proposed GAGSc method, the searching engine is embedded in an evolutionary algorithm and selects candidate genes by mutation and crossover. The proposed LOOLSc method employs a sequential searching algorithm to check every gene for candidate subsets generation.

Sequential forward selection (SFS) and sequential backward elimination (SBE) are two well-known sequential search algorithms. In the context of gene selection, SFS starts with an empty set and sequentially adds a new gene into the set, while SBE starts from a full gene set and sequentially removes one gene. Hence, at each step of search procedure, the gene subset obtained after addition or deletion can lead to the largest improvement in terms classification performance. In this way, the search process can converge toward an optimal gene set. Sequential forward floating selection (SFFS) is an enhanced sequential search algorithm that works similar to SFS, but allows the removal of worst genes during the process of evolving candidate gene set when the classification performance is improved. SFFS initializes the best subset of genes as an empty set and add a new gene at each step. After the best candidate subset is generated, the algorithm searches the genes that can be removed from the best subset until the classification performance can be improved. In this search process, the number of removed genes is not fixed, and depends on the evaluation criterion.

Sequential search algorithms are based on exhaustive search, which result in a high cost in terms of computation time. Comparing with SFS and SBE algorithms, SFFS is more time consuming and is not recommend for analyzing microarray data with a huge number of genes in practice (Sun, Bebis, Yuan, & Louis, 2002). Unlike sequential search, random search does not explore the space of all genes, but randomly selects certain candidate genes. Thus, many useful candidate gene subsets are not taken into account, which leads to the final result being unstable and not highly optimized. Such issues make random search computation costs prohibitive for gene selection in practice. GA search is capable of covering more potential combinations of candidate genes, so that the finally selected informative genes are more reliable, because ideally all potential combinations of genes can be evaluated. In other words, unless the stopping criterion is reached, GA search algorithm will explore the whole space of genes to find better solutions. Such a mechanism assures that GA search usually outperforms sequential and random search. However, GA search is often criticized in literature, because of its huge cost of computational complexity. It generally takes much more time to achieve the final optimized solution than sequential search.

Chapter 4
Experiments

4.1 Datasets

The proposed concept for gene selection is applied to seven well-known benchmark cancer microarray data and one proteomics data. Table 4.1 summarizes the eight datasets used for gene selection in the experiment.

Data name	**Class 1 vs. Class 2**	**Number of Genes**	**Training Samples (class 1/2)**	**Validation Samples**	**Ref.**
Lymphoma	Diffused large B cell lymphoma vs. other types	4026	(42/54) 96	-	1
Leukaemia	ALL vs. AML	7129	(27/11) 38	34	2
CNS Tumour	Survivor vs. Failure	7129	(21/39) 60	-	3
Colon Cancer	Normal vs. Tumour	2000	(22/40) 62	-	4
Ovarian	Cancer vs. Normal	15154	(91/162) 253	-	5
Breast Cancer	Relapse vs. Nor-Relapse	24482	(34/44) 78	19	6
Lung Cancer	MPM vs. ADCA	12533	(16/16) 32	149	7
Esophageal Cancer	None responder vs. Responder	859	(15/12)27	15	8

Table 4.1 Summary of microarray and proteomics datasets used for experiments

1. Lymphoma data (Alizadeh, Eisen, Davis, Ma et al., 2000)

   (available at http://llmpp.nih.gov/lymphoma/)

   This data contains the expression levels of 4026 genes across 96 samples in lymphoma patients. Among them, 42 samples are from "Diffused large B cell lymphoma" group while 54 are from others types.

2. Leukaemia data (Golub, Slonim, Tamayo, Huard et al., 1999)

   (available at http://www.genome.wi.mit.edu/MPR/)

   The biology problem on this data is to distinguish two types of leukaemia, Acute Lymphoblastic Leukaemia (ALL) and Acute Myeloid Leukaemia (AML). The training data contains 38 bone marrow samples (27 ALL and 11 AML). The dimensionality of this microarray data is 7,129 probes from 6817 human genes. The validation testing set consists of 34 samples (20 ALL and 14 AML).

3. CNS cancer data (Pomeroy, Tamayo, Gaasenbeek, Sturla et al., 2002)

   (available at http://www-genome.wi.mit.eud/mpr/CNS/)

   CNS cancer data contains 60 samples, 21 are survivors (class 1) and 39 are failures (class 0). Survivors represent the patients who are alive after the treatment while failures are those who succumb to the central nervous system cancer.

4. Colon cancer data (Alon, Barkai, Notterman, Gish et al., 1999)
   (available at http://microarray. princeton.edu/oncology/)
   Colon cancer data consists of 62 samples collected from colon-cancer patients; 40 samples are labelled as cancer and 22 are labelled as normal. Only 2,000 genes out of total 6,500 genes are selected into the dataset based on the confidence in the measured expression levels.
5. Ovarian data (Petricoin, Ardekani, Ben A Hitt, Fusaro et al., 2002)
   (available at http://clinicalproteomics.steem.com/)
   This dataset contains 253 samples in which 91 samples are healthy while 162 are ovarian cancer. There are total 15,154 proteins for identifying tumor patterns. The raw spectral data of each sample contains the relative amplitude of the intensity at each molecular mass / charge (M/Z) identity.
6. Breast cancer data (van't Veer, Dai, MJ, YD et al., 2002)
   (available at http://www.rii.com/publications/2002/vantveer.htm
   The training data has 78 patient samples: 34 were from the patients who had developed distant metastases within 5 years (labeled as relapse), and the rest 44 samples were from those patients who remained healthy from the disease after their initial diagnosis more than 5 years (labelled as non-relapse). In the testing dataset, there are 12 relapse and 7 non-relapse samples. The dimensionality of this dataset is huge, 24,481 genes that used for discriminating cancer patterns.
7. Lung cancer data (Gordon, Jensen, Hsiao, Hsiaox et al., 2002)
   (available at http://www.chestsurg.org/microarray.htm)
   MPM = malignant pleural mesothelioma, ADCA = adenocarcinoma.
   This data is originally used for classification between malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA) of the lung cancer diagnosis. The complete dataset has 181 tissue samples (31 MPM vs. 150 ADCA). 32 samples are referred as the training set (16 MPM vs. 16 ADCA), and the rest 149 samples are used for testing. Each sample is described by 12,533 genes.
8. Esophageal Cancer data (Hayashida, Honda, Osaka, Hara et al., 2005)
   (available at http://clincancerres.aacrjournals.org/cgi/content/full/11/22/8042/DC1)
   This data contains the expression levels of 859 genes across a training set (27 samples) and a testing set (15 samples). All samples belong to two classes: class 1 – responders (pathologically diagnosed responders to preoperative chemoradiotherapy), and class 0 – nonresponders.

4.2 Unbiased method verification

When analysing microarray data, selection of a data sampling method is important for the verification of final experimental results (Allison, Cui, Page, & Sabripour, 2006; Braga-Neto, Hashimoto, Dougherty, Nguyen et al., 2004), because an improper sampling method often leads to some biased and unreplicapable results (Zhu, Wang, Ma, Rao et al., 2003). For example, Ramaswamy (2003) for breast cancer, and Zhu et al. (2003) for ovarian cancer all claimed that their classification analysis has achieved a very high accuracy (close to 100%). However the experiments are reported unreplicable in the experiments done by other laboratories. Ransohoff (2004) said that these tests are failed to be reproduced due to the process of validation, i.e. sampling method was not well developed.

An unbiased verification scheme has been employed in the experiment to decrease the generalization error in both gene selection and classification stages. This section starts to give a brief review of several popular sampling techniques, then explains the setup of a totally unbiased verification process for all the experiments described in this thesis.

4.2.1 Review of sampling methods

In the machine learning literature, several sampling methods are recognized as an unbiased verification method, such as resubstitution, cross-validation, and bootstrap (Efron, 1979). In this thesis, two major sampling methods, K-fold cross-validation and bootstrap, are discussed here in terms of disadvantages and advantages.

4.2.1.1 Cross-validation

Cross-validation is a sampling technique extensively used in micorarray data analysis (Ambroise, & McLachlan, 2002; Qiu, Xiao, Gordon, & Yakovlev, 2006). According to Ransohoff (2004), cross-validation is "a technique used in multivariable analysis that is intended to reduce the possibility of overfitting and of non-reproducible results. The method involves sequentially leaving out parts of the original sample (‘split-sample’) and conducting a multivariable analysis; the process is repeated until the entire sample has been assessed. The results are combined into a final model that is the product of the training step" (p. 312).

The advantage of cross-validation is that all the data can be used for cross training and testing, and the validation is totally independent to the training process. In the context of microarray data analysis, using cross-validation, the dataset is randomly partitioned into two subsets, training and testing set. Indeed, the goal of implementing cross-validation is to evaluate whether the result is replicable or just caused by chance.

Cross-validation can be generally performed in two ways: K-fold cross-validation and leave-one-out cross-validation (LOOCV). In K-fold cross-validation, samples are randomly divided into K mutually exclusive subsets of approximately equal size. The validation process will be repeated for K rounds, where for each round, K-1 subsets are used for training (*e.g.* classifier training), and the rest one subset for testing. For microarray analysis, 5 or 10 fold is suggested in a typical method cross-validation. (Breiman, & Spector, 1992; Kohavi, 1995).

LOOCV eventually is a K-fold cross-validation, whose number of fold K equals the number of samples (*N*) in given dataset. In LOOCV, all the samples are separated *N* rounds, where for each round, all samples are used for training except one is left for testing. The final result is made by the average performance over *N* testing sets.

For many years, LOOCV has been suggested for evaluating classification performance over the data with a very small number of samples, as it is a nearly unbiased method and works well for estimating bias error, such as the mean squared error. However, Breiman and Spector (1992) have demonstrated that the high variance of leave-one-out cross-validation arises when the prediction rule of the method under verification is unstable. This is mainly because leave-one-out sampling makes the training set very similar to the whole dataset.

4.2.1.2 Bootstrap

Bootstrap, first introduced by Efron (1979), is a new sampling method for small sample size dataset. Empirical studies have showed bootstrap is particularly effective for estimating bias error for very small sample size data, such as microarray data (Braga-Neto et al., 2004; Efron, 1983). Recently, many bootstrap estimators are proposed, in which e0 and the .632 bootstrap are two popular methods and can yield the good results for sampling in classification problems.

The principle of bootstrap method is data sampling with replacement. Suppose a dataset contains only 5 samples labelled as A, B, C, D and E. A sampling with replacement can be described as follows:

1. Randomly draw out one of 5 samples and record its label.
2. Put the sample back to the dataset.
3. Repeat step 1-2 for B times (B is a constant integrator) to have a B labels in a *sequence*.
4. Randomly select a subsequence of 5 labels from the *sequence* obtained in Step 3, and extract the corresponding samples as the *training set* (the first round).
5. Repeat Step 1-4, to construct the *testing set*.

4.2.1.3 Comparison of cross-validation and bootstrap methods

Cross-validation has a disadvantage that the training of model lacks sufficient information due to insufficient observations when the dataset size is too small. Therefore, in the case of partitioning a microarray dataset, cross-validation technique probably increases the risk of overfitting. Critical scientific issues are raised in gene selection literature by using cross-validation for generalization error estimation (Braga-Neto et al., 2004). Nevertheless, cross-validation is still considered a robust and unbiased technique in microarray data analysis, if it is well designed and organized in experiments (Asyali, Colak, Demirkaya, & Inan, 2006).

Bootstrap uses a replacement resampling approach, and constructs training and testing sets with the exact same size as the whole dataset, while for cross-validation, both training and testing sets use only a subset of the whole dataset. Thus, bootstrap method has an advantage of modelling the impacts of the actual sample size (Fan & Wang, 1996). The disadvantage is, bootstrap method yields a good result only after hundreds of iterations, which makes it more expensive than cross-validation in terms of computational complexity.

4.2.2 The totally unbiased verification scheme

A typical microarray data analysis usually includes two procedures, gene selection and disease diagnosis (microarray classification). In the first stage, the main goal is to identify the biomarker genes that can efficiently represent the unique characteristics of the given microarray data. In the second stage, the target is to construct an efficient classifier for disease diagnosis using the microarray data with the selected biomarker genes. Proper data sampling methods are fundamental to avoid the generalization error in both gene selection and classification stages.

In most previous microarray data analysis work, sampling method is employed mainly for classification procedure, but not for gene selection procedure (Huerta, Duval, & Hao, 2006; Li, & Xiong, 2002). Such mechanism makes the classification results eventually with bias, because the informative genes are selected from the whole dataset and not well estimated in terms of the generalization error. In practice, testing data is blind in real biology experiments thus is not allowed to be included in either gene selection or classification modelling. Therefore, the bias occurring in gene selection procedure may finally result in an unreplicable disease diagnosis performance.

A totally unbiased verification policy for microarray analysis should guarantee that no generalization error occurs in either gene selection or classification procedures. To this end, efficient data sampling method should be used in the two procedures to maximally decrease the generalization error. In other words, the reliability and generalizability of the informative genes selected in gene selection stage should be evaluated on certain independent testing subsets, and then these genes can be used for classification. The classification also needs to employ the verification methods to estimate the bias error. Such procedure can be summarized as a totally unbiased verification scheme in case b of Fig. 4.1.

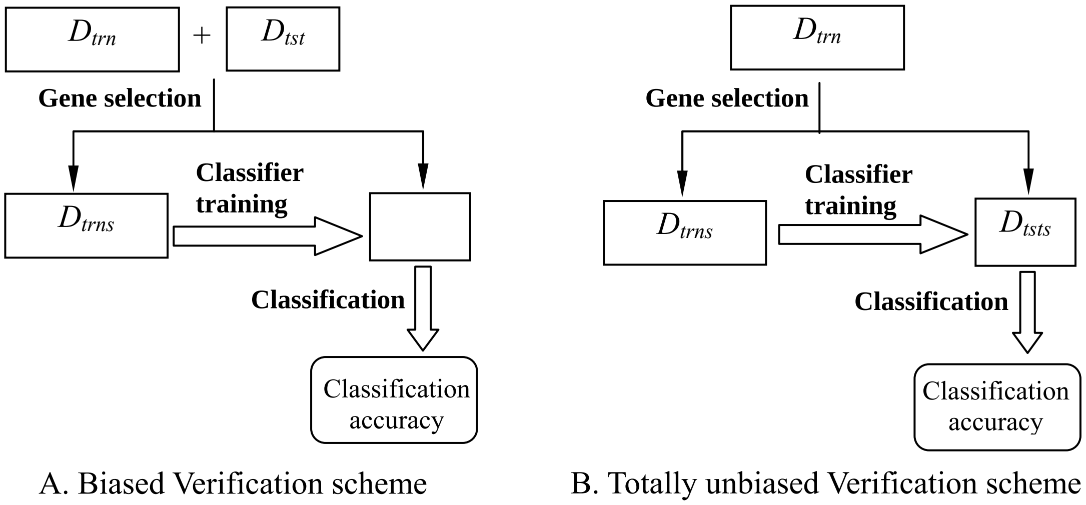


Fig. 4.1 The comparison between a biased and a totally unbiased verification scheme, where D_{trn} and D_{tst} are the training and testing set, D_{trns} and D_{tsts} are the training and testing set with selected genes, respectively. In case A (biased verification scheme), the testing set is used twice in gene selection and classifier procedures, which creates a bias error in the final classification results. Whereas in case B (the totally unbiased scheme), the testing set is only used in classification stage, so that it is independent in gene selection and classifier training procedures.

4.3 Experiment Setup

4.3.1 Software and hardware

The experiments are implemented into the Matlab environment on two computers with 3.2 GHz Pentium 4 and 2048 MB RAM. Relevant software used for comparison and gene selection modelling in the experiments is described in Table 4.2:

Software/Algorithm	Note	Availability
NeuCom	A Neuro-computing decision support system	www.theneucom.com
T-test algorithm	For gene selection and classifier training (Section 4.4)	Matlab toolbox
SNR algorithm	For gene selection and classifier training (Section4.4)	www.theneucom.com
KNN algorithm	For microarray dataset classification (Section 4.4 ~ 4.5)	Matlab V7.0 toolbox
SVM algorithm	For microarray dataset classification ((Section 4.5)	www.theneucom.com

Table 4.2 Relevant software used for gene selection comparison and modelling

4.3.2 Microarray diagnosis setup

As suggested in literature for estimating generalization error (Breiman et al., 1992; Kohavi, 1995), a 5-fold cross-validation schema is applied to all datasets except on those datasets with the training and testing set originally separated. For each cross validation, a totally unbiased verification scheme as in Fig. 4.1 is used, where both gene selection and classification are working only on the training set, so that no testing information is included in any part of the cancer diagnosis modeling.

4.3.3 Parameters setup in the proposed consistency methods

For consistency evaluation, the dataset is randomly partitioned into two subsets. One subset contains 2/3 of all samples, and the other subset has the rest 1/3 samples. Using a classifier such as KNN or SVM, two classification accuracies can be computed on two subsets, respectively, the absolute difference between these two accuracies is defined as the consistency (C) in terms of classification performance (refer to Eq. 3.5). After several hundred iterations, the mean value of computed consistency is taken as the final result.

4.3.4 Parameters setup for relevant algorithms

The parameters setting for the published two gene selection algorithms and two traditional classifiers are summarized as follows:

1. T-test gene selection algorithm:

α: 0.05 (α is the level for calculating the confidence interval).

Test type: 1-tail.

2. SNR gene selection algorithm:

Test type: test and difference in genes between classes.

3. KNN classifier:

K: 1 (K – the number of nearest neighbours considered).

4. SVM classifier:

Type of kernel: Linear kernel.

For comparison of T-test, SNR and DA on gene selection, the number of top-ranked genes to be selected by statistical function in Eq. (2.25) is set as 50. Fewer genes to be selected may make the operation unreliable, whereas too many genes might introduce noise to the experiment and make the optimization become very time costly. Previous studies indicate that a few dozen to a few hundred top-ranked genes can efficiently classify the different disease patterns in most microarray experiments (Li, & Yang, 2002).

4.3.5 Parameters setup for proposed GAGSc method

In GAGSc method, all genes of a given microarray dataset (the search space) are first segmented into ρ segments (refer to Fig. 3.6), and ρ is set as 20. Detailed discussion of this parameter is in chapter 5. For each fold dataset obtained from cross-validation data sampling, GA will run N generations (refer to section 3.5.3) in the training process to find the informative genes, and N is set as 100.

There are two options for choosing resampling times B (refer Section 3.5.3) in every computing consistency procedure depending on how many samples are in the dataset, one is 50 for those datasets with more than 30 samples, and the other is 30 for the datasets with less than 30 samples.

Additionally, as the number of genes is huge, so that for each mutation or crossover operation, the permutation of candidate genes (new generation of chromosome) tends to be massive and GA evaluation becomes extremely time costly. Hence, a parameter denoted as T is introduced to the stopping criterion (refer to section 3.5.3) for GA program, which makes the operation of mutation or crossover must be completed within T seconds. In the experiment, T is set as 240 seconds.

For clarity, all the parameters for GAGSc method are summarized in Table 4.3.

Parameters	Value
ρ - number of initial selected genes	*20*
N - number of generations	100

B - number of resampling times	if sample size >= 30 ***B*** = 50;
	else ***B*** = 30.

T – Time limitation for GA operation (mutation or crossover)	240s

Table 4.3 Parameters setup in GAGSc

4.3.6 Parameters setup for proposed LOOLSc and LOOLS methods

There are only two inputting parameters in LOOLSc and LOOLS methods, one is a pre-defined number of genes to be selected that is denoted as *m*, and the other is the number of resampling times that has been discussed in above section.

Because there is no agreement on how many genes are best for the classification on different dataset, and LOOLSc method is very costly in terms of computation complexity, *m* is set to 30 in the experiments. During the LOOLSc gene selection process, all classification accuracies obtained using different number of selected genes (from 1 to 30) can be recorded for comparison.

For clarity, the two parameters for LOOLSc and LOOLS methods are listed in Table 4.4.

Parameters	Value
m – a pre-defined number of genes to be selected	50

B - number of resampling times	if sample size >= 30 ***B*** = 50;
	else ***B*** = 30.

Table 4.4 Parameters setup in LOOLSc and LOOLS

4.4 Consistency concept verification

There are two objectives for the experiments of consistency concept verification. The first objective is to estimate the consistency of eight microarray datasets. To reduce the computational complexity, a simple classical t-test algorithm is used for finding informative genes. Then, KNN classifier is employed to evaluate the consistency in terms of Eq. (3.2).

The second objective is to investigate the consistency concept on three different gene selection methods (t-test, SNR and DA). For a given dataset, a set of informative genes are first selected using the above three algorithms. Then, the selected genes are used for classification on the dataset in NeuCom (KEDRI, 2002) environment. In this experiment, four datasets: leukaemia, lymphoma, CNS cancer, and colon data, are used for comparison.

4.4.1 Estimating consistency on datasets

Table 4.5 summarizes the consistency of eight microarray datasets. As seen in the table, there is significant difference in the consistency among these eight datasets. The consistency value ranges from 0 to 1, in which a smaller value reflects a better consistent characteristic of dataset. Lung cancer dataset has the best consistency (0), which means there is no difference between the classification of training set and that of testing set in terms of performance. In contrast, esophageal cancer data has the worst consistency value (0.3928) on training set, and 0.4670 on testing set.

Data	Consistency value
Lymphoma	0.0450
Leukaemia	0.0988/0.1238
CNS Tumour	0.3482
Colon Cancer	0.2640
Ovarian	0.1589
Breast Cancer	0.2219/0.3400
Lung Cancer	0
Esophageal Cancer	0.3928/0.4670

Table 4.5 Consistency comparison on eight datasets. Note that for those microarray datasets with training and testing set originally separated, such as Leukemia, Breast cancer and Esophageal cancer datasets, consistency is calculated on the training and testing sets respectively, e.g. the consistency of leukemia on training set is 0.0988, and the consistency on testing set is 0.1238. Here, t-test is used for identifying informative genes, and KNN is used for evaluating the consistency through Eq. (3.2).

4.4.2 Investigating consistency on gene selection

This experiment focuses on evaluating the consistency capability of previous gene section methods, t-test, SNR and DA, where DA is a method with a consistency concept (see section 2.4). The DA gene selection method is used to compare the other two methods in terms of classification accuracy, with the purpose of investigating the effectiveness of consistency utilized in gene selection.

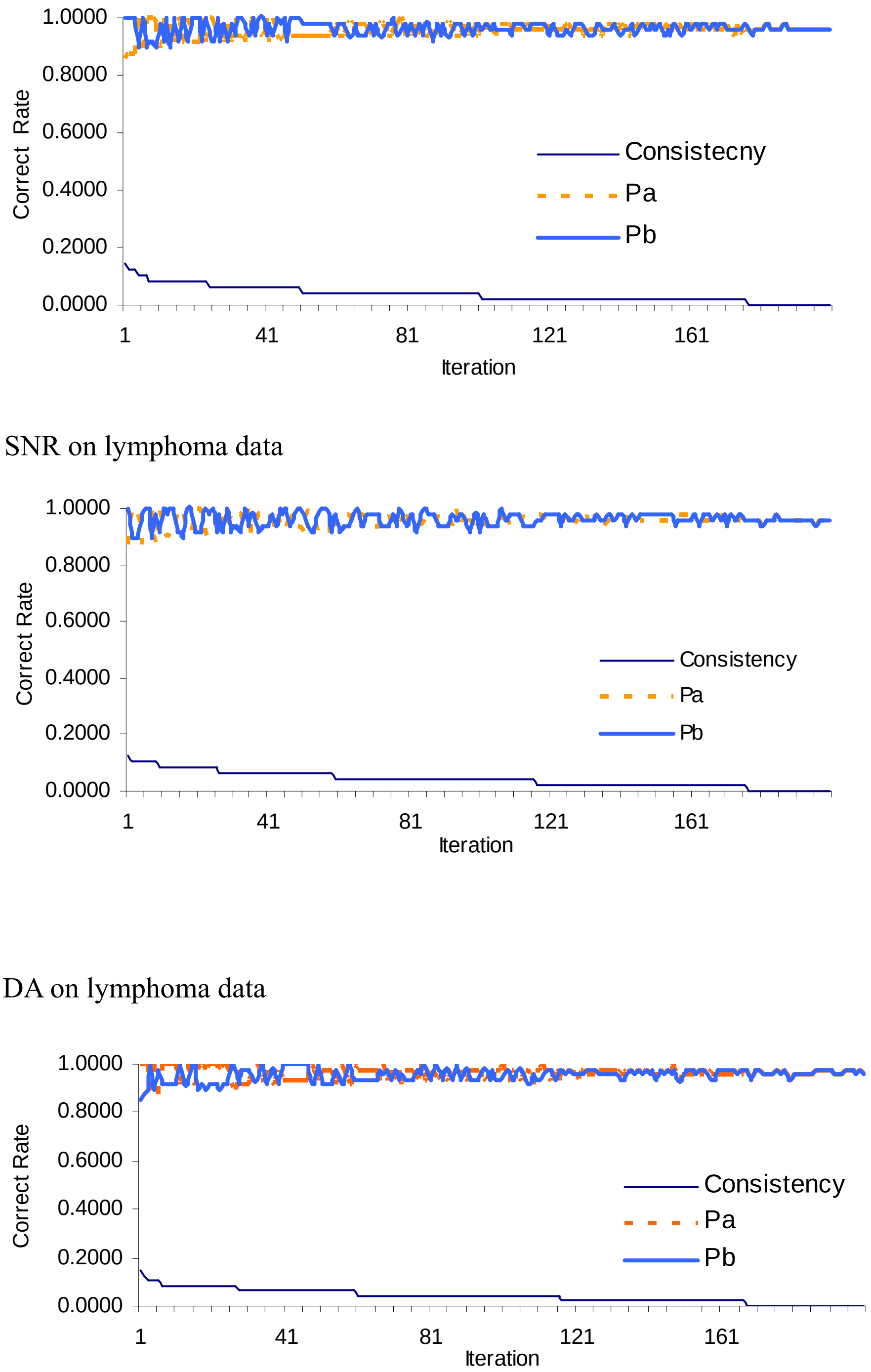
Table 4.6 summarizes the evaluated consistency results from DA method together with two gene selection methods based on T-test and SNR algorithms on leukaemia, colon, lymphoma and CNS cancer datasets. The consistency here is evaluated by Eq. (2.23), and the results are represented by the mean value of the consistencies calculated for 200 iterations.

Data Consistency Method	**Leukaemia**	**Colon**	**Lymphoma**	**CNS**
t-test	0.0529 ± 0.0400	0.1227 ± 0.0916	0.0411 ± 0.0314	0.0863 ± 0.0710
SNR	0.0455 ± 0.0383	0.1123 ± 0.0853	0.0421 ± 0.0299	0.0985 ± 0.0744
DA	0.0674 ± 0.0581	0.1232 ± 0.0943	0.0379 ± 0.0249	0.1275 ± 0.0942

Table 4.6 Consistencies evaluated by three gene selection methods, where each consistency is represented as an average value ± standard deviation. The number of iteration times is 200.

As shown in Table 4.6, different datasets have very different inherent consistency values. The consistency of Lymphoma and Leukaemia is significantly better than that of Colon and CNS data. In contrast, Colon data has the highest inconsistency that is nearly three times higher than the most consistent dataset (Lymphoma). In addition, the consistency is seen also varying over different gene selection methods. SNR method outperforms the other methods on three of the four datasets, while Data-adaptive method wins in the fourth dataset. Among these datasets, the best consistency occurs when Data-adaptive method is used for gene selection on Lymphoma dataset.

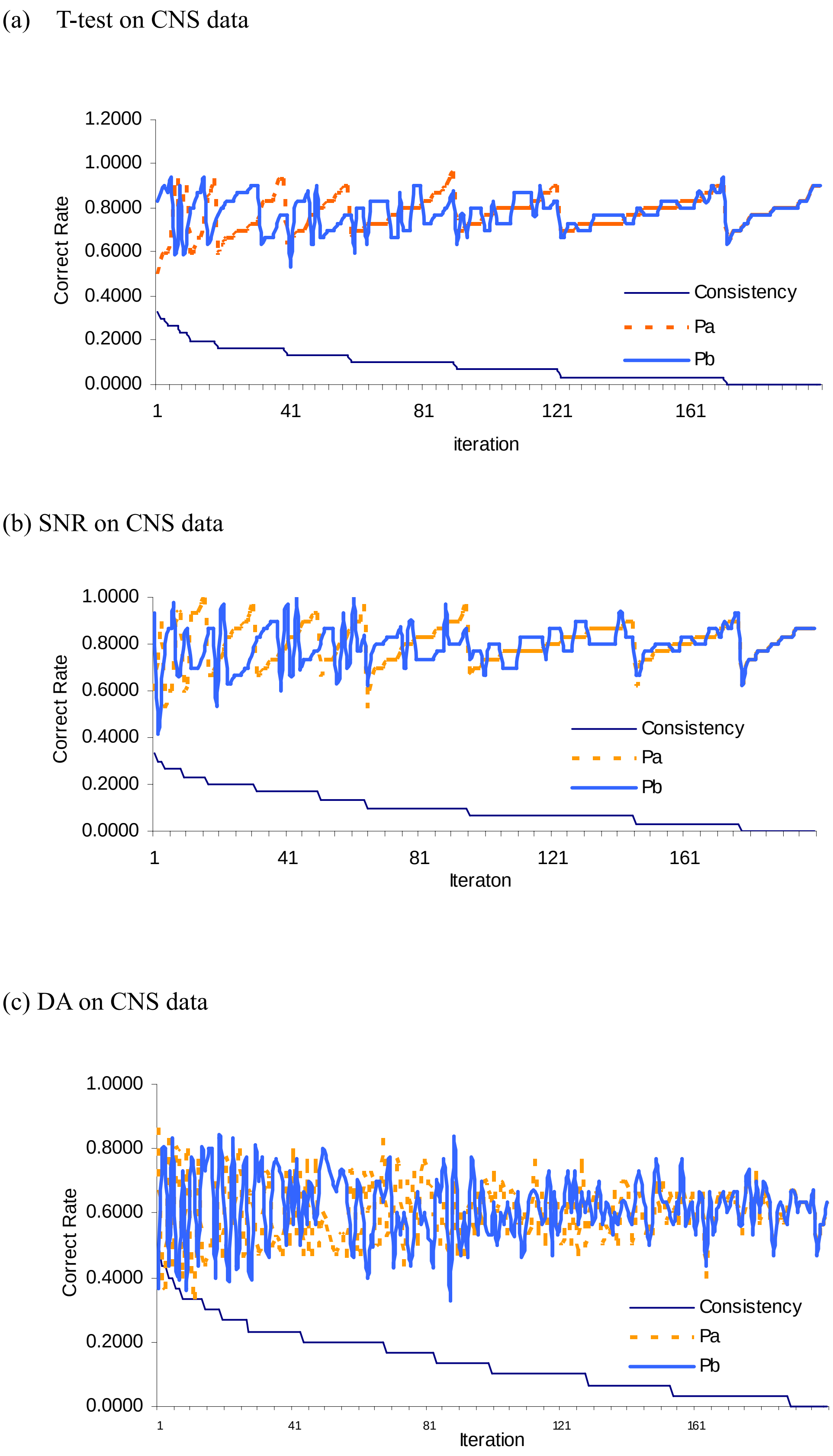
Fig. 4.2 shows that for all three gene selection methods, the consistency of Lymphoma dataset in terms of performance is good and varies slightly, which ranges between 0.1 and 0. Correspondingly, it displays better average classification performance (P_a and P_b) of Lymphoma data. In contrast, in Fig. 4.3 (a)–(c), CNS data shows more variance in both consistency and performance. Interestingly, it is clear that better consistency is related to the high performance. In Fig. 4.3, high variance is shown in the value of consistency, P_a and P_b using DA method, while T-test and SNR methods show less variance.


(a) T-test on lymphoma data

(b) SNR on lymphoma data

(c) DA on lymphoma data

Fig. 4.2 Comparison of three gene selection methods (T-test, SNR and DA) on Lymphoma data. Horizontal axis represents the iterations of classification tests using selected informative genes. Vertical axis represents the value of consistency via Eq. (3.2), classification accuracy of D_a and D_b (P_a and P_b) (calculated by Eq. (3.3)), respectively.



(a) T-test on CNS data

(b) SNR on CNS data

(c) DA on CNS data

Fig. 4.3 Comparison of three gene selection methods (t-test, SNR and DA) on CNS cancer data. Horizontal axis represents the iterations of classification tests using selected informative genes. Vertical axis represents the value of consistency obtained by Eq. (3.2), classification accuracy P_a and P_b (see Eq. (3.3)), respectively.

The above experimental results show that the consistency issue exists obviously in most microarray data analysis. Different datasets may have different inherent consistency characteristics, and different gene selection methods are embedded with different consistency capability on microarray data analysis. Additionally, another remarkable finding is that a better classification accuracy can be easier to achieve on a dataset with high consistent characteristic, or by a method with high consistency capability.

4.5 Verification of proposed consistency methods

Two experiments are presented in this section to verify the proposed consistency based gene selection methods: GAGSc and LOOLSc. The first experiment presents GAGSc experimented with eight cancer microarray datasets (see section 4.1), and compared with the well-known reported experimental results of these datasets in terms of the cancer diagnosis prediction accuracy. Note that these reported results can be found in the original papers of eight cancer microarray datasets (refer to the cited papers in section 4.1).

In the second experiment, LOOLSc method is compared with LOOLS method (the similar method as LOOLSc based on the same LOOE criterion, but without consistency measurement) to evaluate the effectiveness of using consistency concept in gene selection procedure.

4.5.1 GAGSc method

Table 4.7 ~ 4.14 show the classification results of GAGSc on the independent validation set of seven benchmark microarray datasets and one proteomics set (refer to section 4.1). In these tables, TP, TN, FP and FN represents true positive, true negative, false positive and false negative of the confusion matrix, respectively, which are commonly used for describing the dispositions of the dataset of instances (Fawcett, 2003).

Fig. 4.4 ~ 4.11 present the optimizing results obtained by GAGSc method on eight microarray datasets. In these figures, the horizontal axis represents the number of GAGSc optimizing iterations, and the vertical axis represents the resulted consistency *C* from Eq. (3.2), classification accuracy *P* on the testing set and ratio *R* of Eq. (3.15), respectively. Note that accuracy *P* is not the final calculated accuracy on the independent validation set, but the accuracy on one subset for calculating consistency in the gene selection procedure.(refer to Fig. 3.2 ~ 3.3)

4.5.1.1 Lymphoma data

Table 4.7 shows the classification results of GAGSc on Lymphoma data, and Fig. 4.4 illustrates the GA optimizing procedure of GAGSc in 5-fold cross-validation, where consistency and classification accuracies are recorded at every optimizing step.

As shown in table 4.7, the overall classification accuracy on the testing set of Lymphoma dataset is fairly high (greater than 95%). The number of selected informative genes is around 30, and the final calculated classification accuracy is stable (94.74% ~ 100%). Moreover, the results of confusion matrix (TP, TN, FP and FN) have shown that the proposed GAGSc method is very effective on Lymphoma dataset in terms of both classification accuracy (TP and TN) and misclassification rate (FP and FN).

Lymphoma data	Number of selected genes	TP	TN	FP	FN	Classification accuracy
Fold1	36	8	10	0	1	94.74%
Fold2	25	12	6	0	1	94.74%
Fold3	34	11	7	1	0	94.74%
Fold4	36	10	9	0	0	100%
Fold5	32	10	9	0	1	95.00%
Overall classification accuracy: **95.84%**						

Table 4.7 The classification validation results of GAGSc method on Lymphoma data. Note that 5-fold cross-validation is used for calculating classification accuracy. TP – True positive, TN – True negative, FP – False positive, and FN – False negative.

Fig. 4.4 gives the optimizing procedure of the proposed GAGSc gene selection. The optimized consistency is seen being decreased to below 0.1, meanwhile the training classification accuracy is increased to above 90%. It shows that the proposed GAGSc algorithm is capable of improving consistency in a GA optimizing process. (A smaller the consistency value indicates a better consistent characteristic of data).

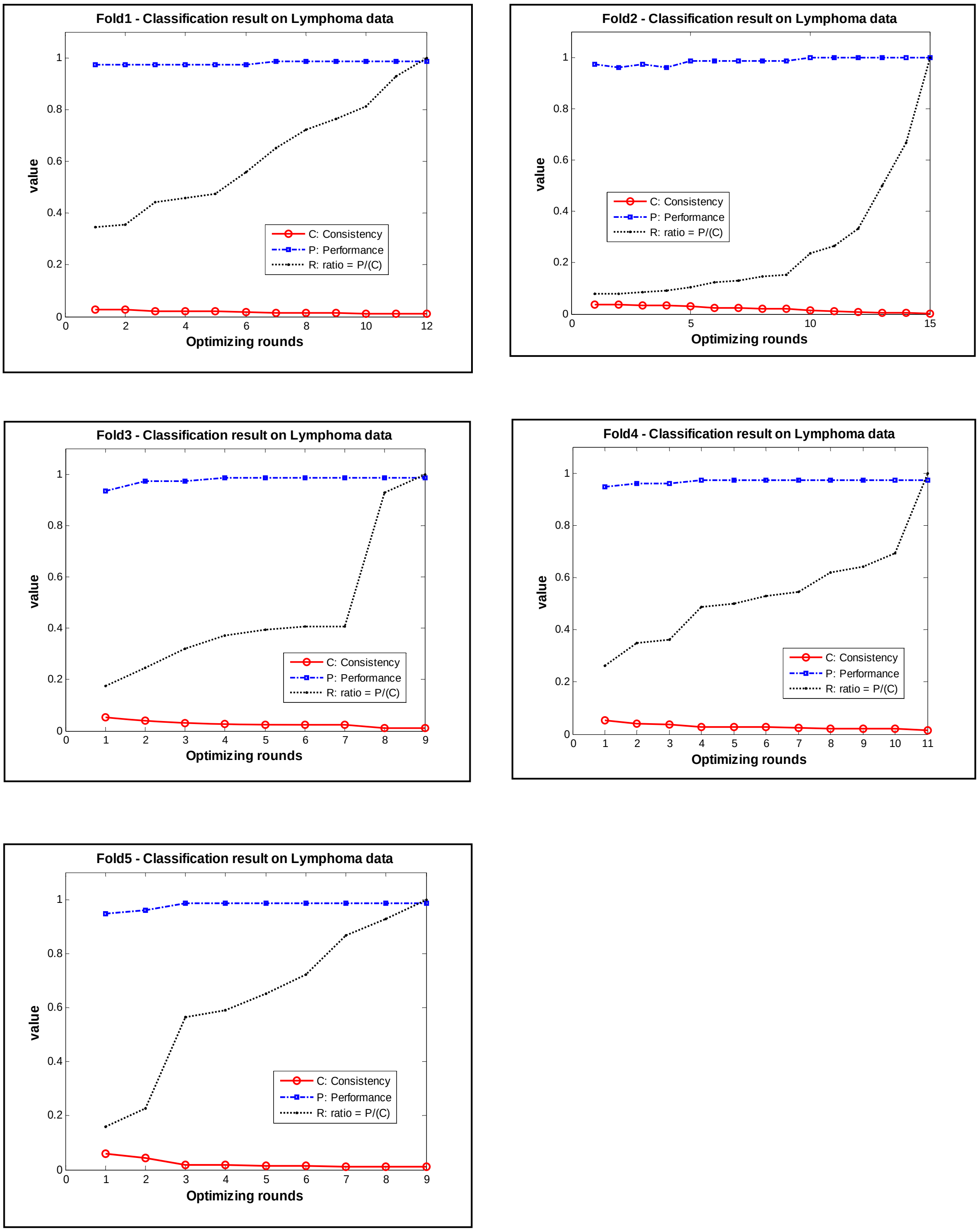


Fig 4.4 The optimizing results of GAGSc on Lymphoma data, where horizontal axis represents the optimizing rounds, and vertical axis shows the results of consistency (C), classification performance (P) and the ratio (*R*) (Eq. 3.15) to consistency and performance calculated in the optimizing process. Note that accuracy *P* is the training classification accuracy obtained in the classifier optimizing process.

4.5.1.2 Leukaemia data

Table 4.8 and Fig. 4.5 present the classification and consistency results of GSGAc and optimizing process on Leukaemia data, respectively. Table 4.8 shows that the achieved classification accuracy on the testing set is about 95%, when 35 GSGAc genes are used for constructing the final optimized classifier. In Fig. 4.5, after 15 rounds optimization based on the improvement of ratio R to consistency and classification performance (refer to Eq. 3.15), the classification accuracy on the training set is improved to 1 and the consistency value is reduced to 0, indicating that the maximum consistency is obtained.

Microarray dataset	Number of selected genes	TP	TN	FP	FN	Classification accuracy
Leukaemia data	35	12	20	0	2	94.12%

Table 4.8 The classification validation result of GAGSc method on Leukaemia data

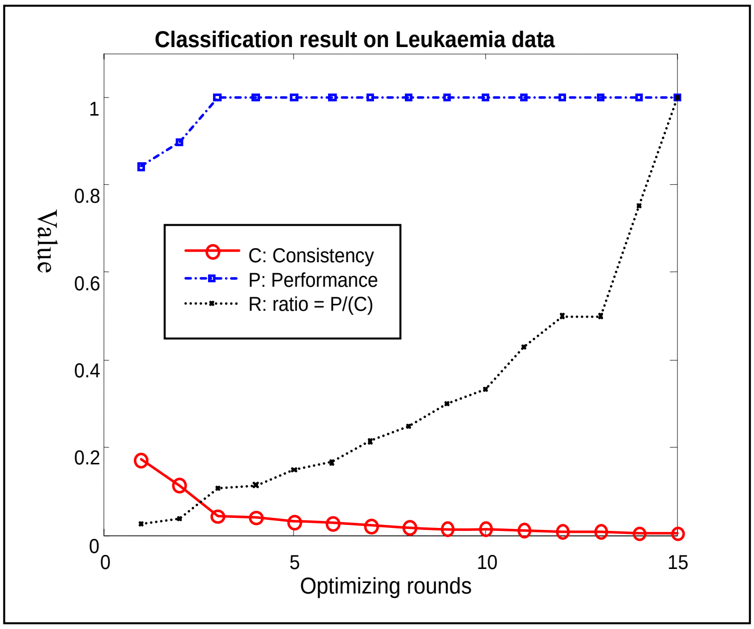


Fig 4.5 The results of GAGSc optimization on Leukaemia data

4.5.1.3 CNS cancer data

Table 4.9 and Fig. 4.6 present the experimental results obtained by GAGSc method on CNS cancer data. Table 4.9 shows that the classification results on 5 folds of CNS cancer dataset have high variance, which the highest accuracy is 83.33% while the lowest is only 41.67%. The overall accuracy is only 65%, which is not acceptable for the real clinical problem of disease diagnosis. The confusion matrix clearly shows that one misclassification rate (FN) is high, e.g. FN obtained on fold2 and fold3 are 5 that is larger than the accuracy rate (TN), so that the classification accuracies on fold2 and fold3 are very low.

CNS cancer data	Number of selected genes	TP	TN	FP	FN	Classification accuracy
Fold1	44	9	1	2	0	83.33%
Fold2	56	4	3	0	5	58.33%
Fold3	43	3	2	2	5	41.67%
Fold4	44	7	2	3	0	75.00%
Fold5	44	6	2	4	0	66.67%
Overall accuracy: **65.00%**						

Table 4.9 The classification validation results of GAGSc method on CNS cancer data.

Fig. 4.6 shows that the consistency value of CNS cancer dataset is quite high (around 0.4) and cannot be successfully decreased in the optimizing process. The classification accuracy on training sets on four folds data rises approximately from 60% to 80%, meanwhile the consistency is decreased from 0.4 to 0.2. Although the accuracy on the rest one fold data is significantly improved, from 40% to 80%, the best consistency is still greater than 0.2, which means the consistency is not satisfying and the effectiveness of utilization GAGSc method on this dataset is limited. Such a situation results in the bad overall classification accuracy (65.00%) on an independent testing set.

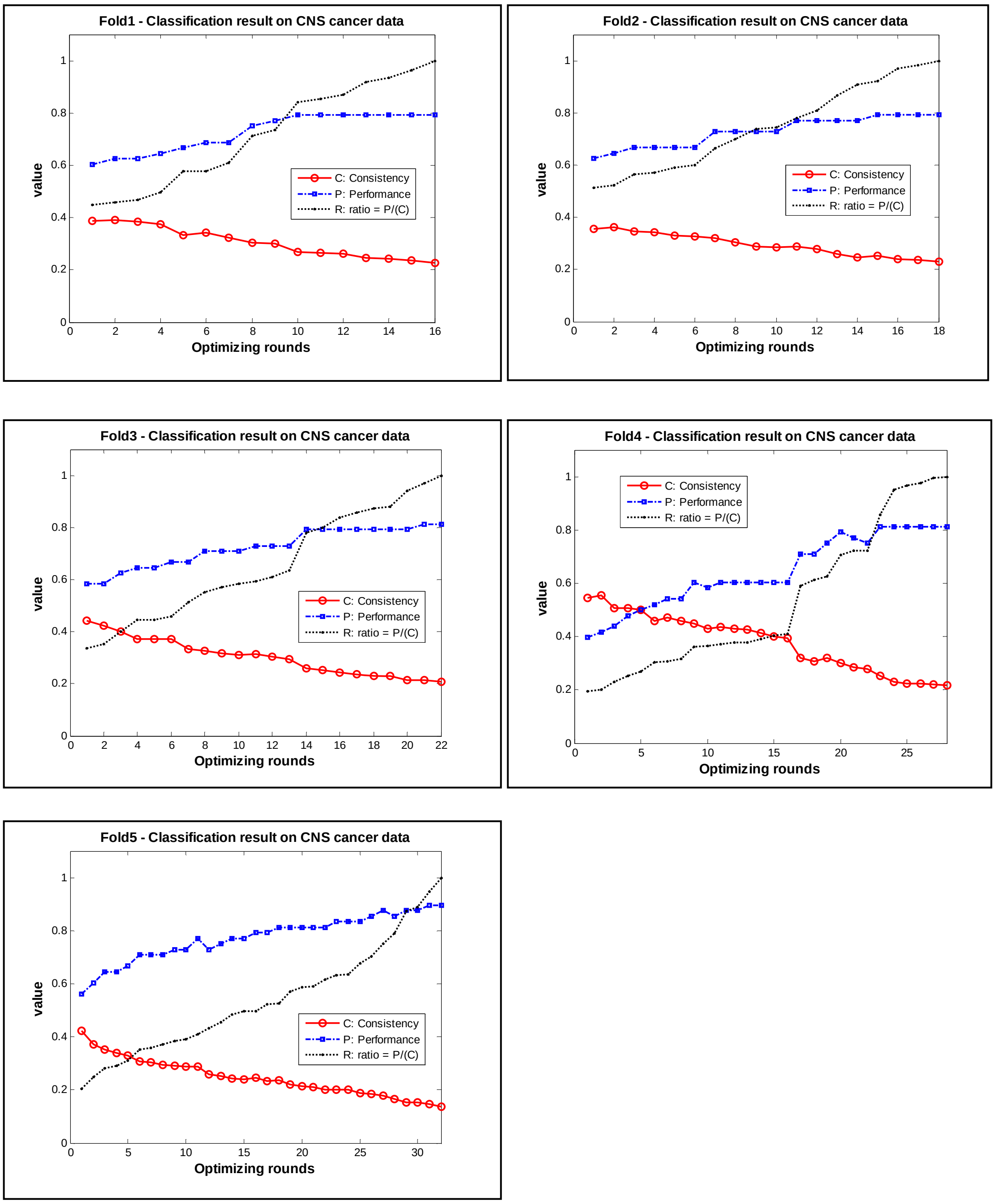


Fig. 4.6 The results of GAGSc optimization on CNS cancer data.

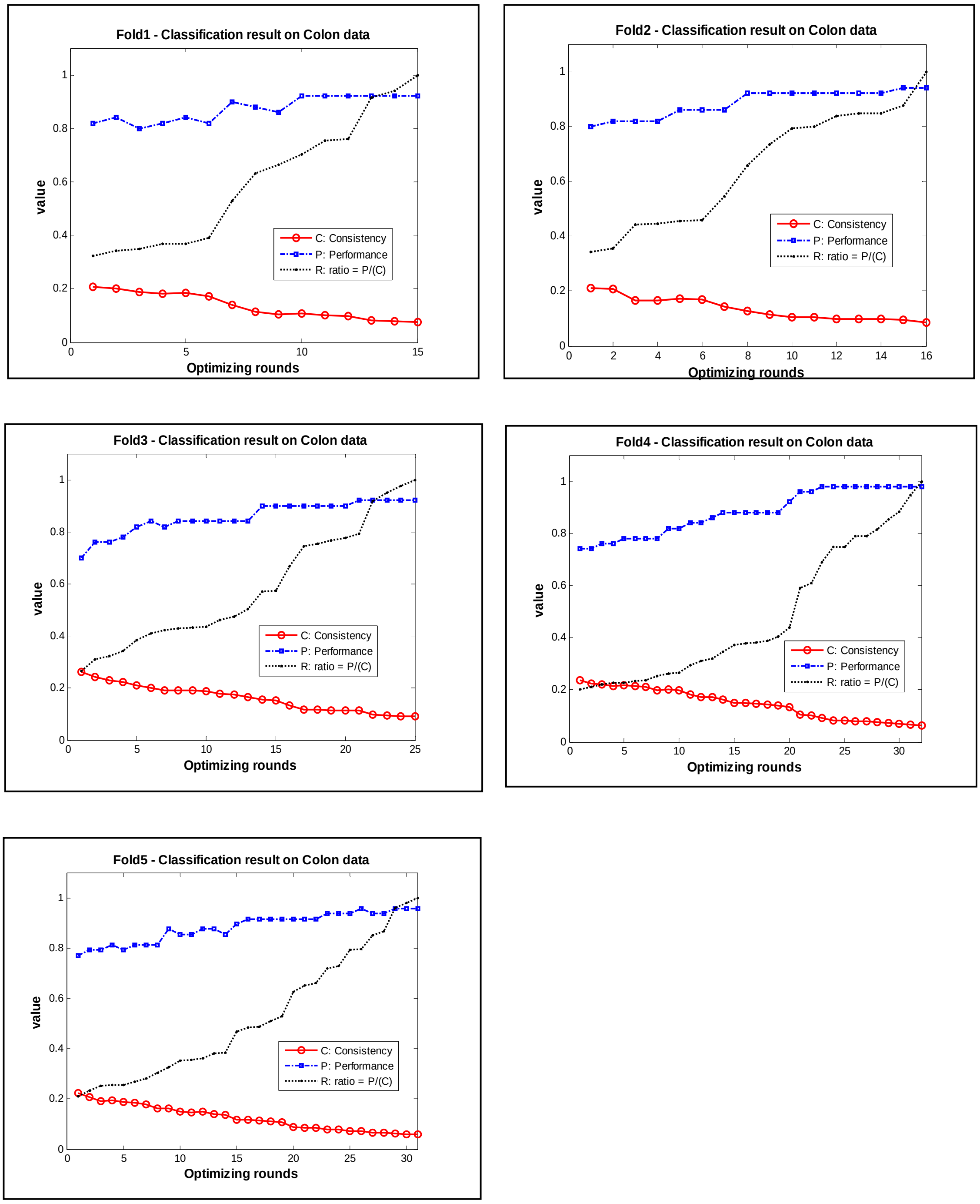
4.5.1.4 Colon data

Table 4.10 and Fig. 4.7 show the experimental results obtained by GAGSc method on Colon cancer data. As presented in Table 4.10, the highest classification accuracy (91.67%) is obtained on fold 1 and fold 4 data in the classifier optimizing process, while the lowest one (66.67%) appears on fold 3. The diffidence between these computed classification accuracies is large, which shows Colon dataset has a relatively high variance of consistency characteristic. The final selected informative genes are more than 22, and the overall classification accuracy is lower than 85%.

Colon data	Number of selected genes	TP	TN	FP	FN	Classification accuracy
Fold1	22	4	7	0	1	91.67%
Fold2	17	4	6	2	0	83.33%
Fold3	21	2	6	1	3	66.67%
Fold4	29	5	6	1	0	91.67%
Fold5	28	1	11	0	2	85.71%
Overall accuracy: **83.81%**						

Table 4.10 The classification validation results of GAGSc method on Colon data.

Fig. 4.7 shows that the consistency and performance are improved significantly. For example, in fold 1, the classification accuracy rises approximately 10% (from 80% to 90%) coupled with the improvement of consistency (from 0.2 to 0.1). The improvement of classification performance obtained on 5 folds data is different, which the performance on fold 3 - 5 is improved more significantly than that on fold 1- 2. Meanwhile, the optimizing rounds are also different, which the classifier optimized over 25 times in the cases of fold 3 - 5. On the contrary, in fold 1-2, the classifier is optimized less than 20 rounds.



Fold1 - Classification result on Colon data

Fold2 - Classification result on Colon data

Fold3 - Classification result on Colon data

Fold4 - Classification result on Colon data

Fold5 - Classification result on Colon data

Fig. 4.7 The results of GAGSc optimization on Colon data

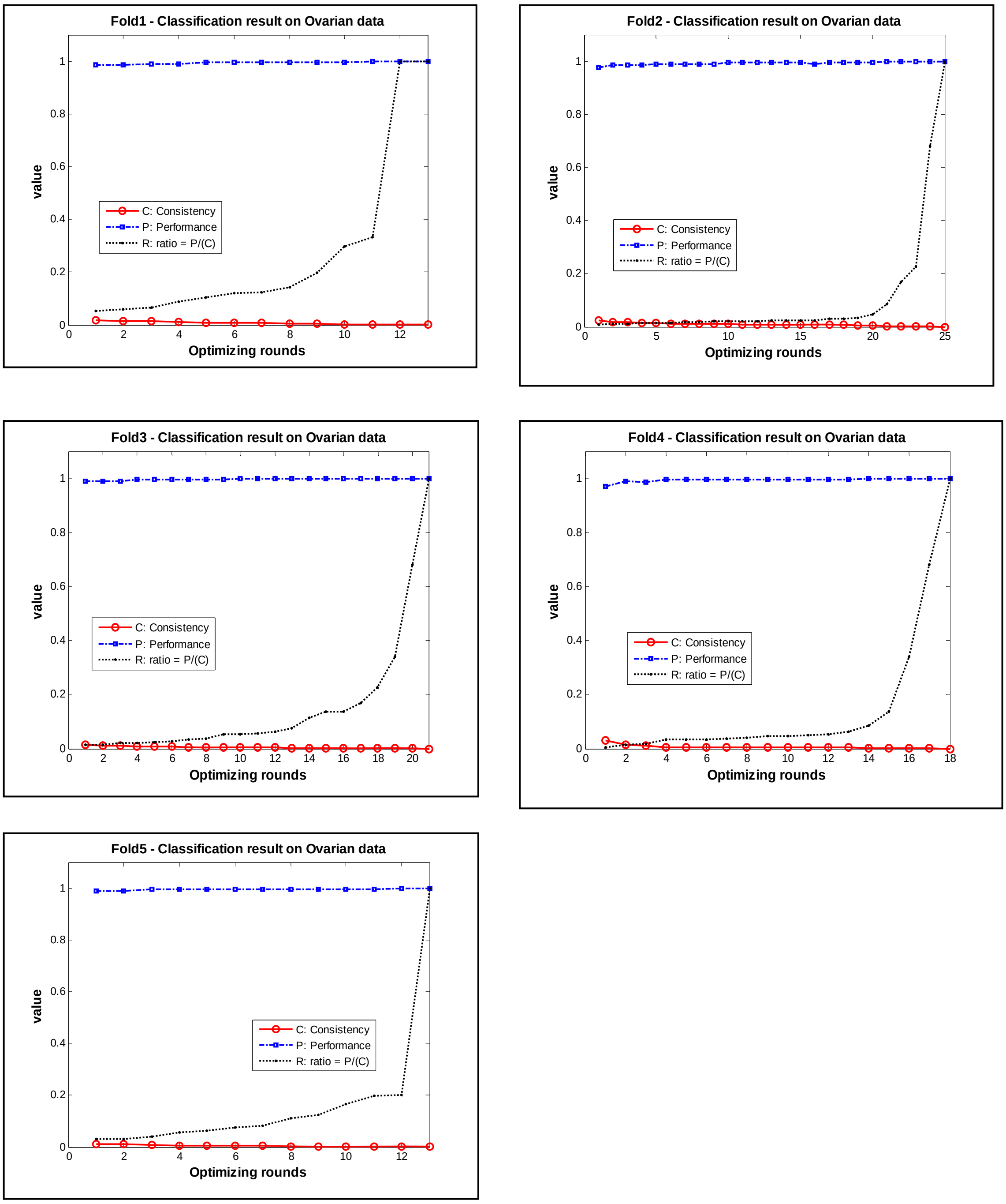
4.5.1.5 Ovarian data

Table 4.11 and Fig. 4.8 give the experimental results obtained by GAGSc method on ovarian cancer dataset. Table 4.11 shows the classification results based on the informative genes selected by GAGSc method. The proposed GAGSc method produces an overall accuracy is of 98.80%. The difference between the highest accuracy (100%) and the lowest accuracy (98%) is only 2%. Moreover, the confusion matrix shows both the classification accuracy rate and misclassification rate are very good, *e.g.* there is no samples are misclassified in the cases of fold4 and fold5.

Ovarian data	Number of selected genes	TP	TN	FP	FN	Classification accuracy
Fold1	18	25	24	0	1	98.00%
Fold2	28	31	18	1	0	98.00%
Fold3	24	33	16	1	0	98.00%
Fold4	24	34	16	0	0	100%
Fold5	34	38	15	0	0	100%
Overall accuracy	**98.80%**					

Table 4.11 The classification validation results of GAGSc method on Ovarian data.

Fig. 4.8 shows that both the classification performance and consistency is stable during the process of classifier optimization. It turns out that the ovarian dataset has a good and low-variant consistency characteristic, which results in the successful classification results on the independent testing set of each fold data. Consequently, the improvement of consistency is less than 0.05 in the cases of 5 folds.



Fold1 - Classification result on Ovarian data

Fold2 - Classification result on Ovarian data

Fold3 - Classification result on Ovarian data

Fold4 - Classification result on Ovarian data

Fold5 - Classification result on Ovarian data

Fig. 4.8 The results of GAGSc optimization on Ovarian data.

4.5.1.6 Breast cancer data

Table 4.12 and Fig. 4.9 present the experimental results of breast cancer dataset. Table 4.12 shows that the low classification accuracy on the testing set is in relation to the high inconsistency characteristic of breast cancer dataset. The classification accuracy obtained by GAGSc method with 50 selected informative genes is only 63.16%, which is not practically useful for identifying disease patterns in real clinical area.

Microarray dataset	Number of selected genes	TP	TN	FP	FN	Classification accuracy
Breast cancer data	50	5	7	5	2	63.16%

Table 4.12 The classification validation results of GAGSc method on Breast cancer data.

Fig. 4.9 presents the bad consistency and classification accuracy obtained by GAGSc method in the optimizing process. The best classification accuracy on the training data in gene selection procedure is 80%, when the final optimized consistency (approximately 0.2) is achieved after 9 iterations. Additionally, there are only 9 optimizing rounds in the classifier training process, which indicates the classifier is difficult to be optimized for breast cancer dataset respect using the proposed consistency concept.

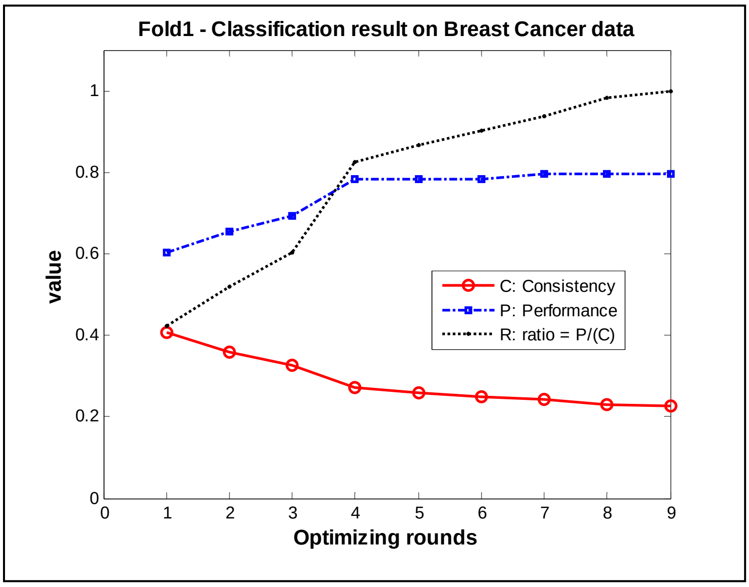


Fig. 4.9 The optimizing results of GAGSc method on Breast cancer data.

4.5.1.7 Lung cancer data

Table 4.13 and Fig. 4.10 present the results obtained by GAGSc method on lung cancer data. As shown in Table 4.13, the experiment result of Lung cancer data reaches a satisfactory level in which the classification accuracy on testing set is 91.28% with 34 selected genes identified by GAGSc method.

Microarray dataset	Number of selected genes	TP	TN	FP	FN	Classification accuracy
Lung cancer data	34	121	15	0	13	91.28%

Table 4.13 The classification results of GAGSc method on Lung cancer data.

As shown in Fig.4.10, the classifier is only optimized 9 times. Unlike the bad consistency and classification performance in the Breast cancer dataset, the difficulty in the optimizing process here is due to the inherent consistency characteristic of lung cancer dataset. It can be seen that the initial classification accuracy is greater than 90%, and the consistency calculated in the first round is about 0.1, so that it only takes 9 optimizing rounds to achieve a high classification accuracy coupled with a good consistency in the training process.

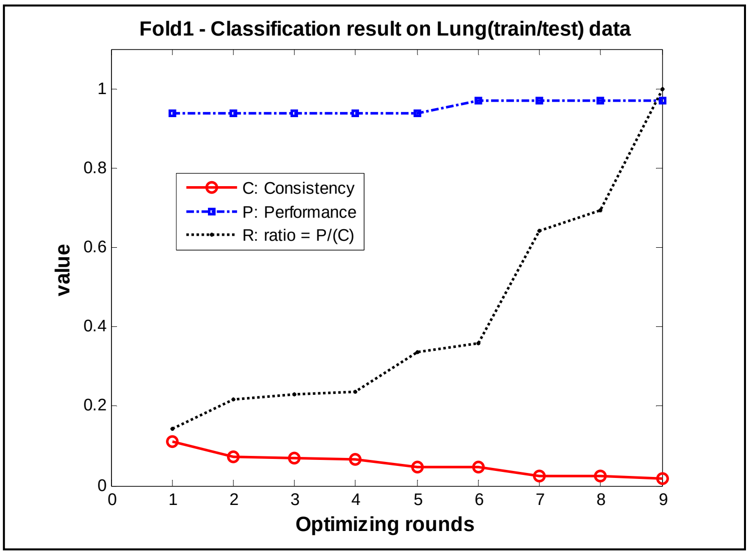


Fig. 4.10 The results of GAGSc optimization on Lung cancer data.

4.5.1.8 Esophageal cancer data

Table 4.14 and Fig. 4.11 show the experimental results obtained by GAGSc method on Esophageal cancer data. As presented in Table 4.14, the final classification performance of Esophageal dataset is unsuccessful, because the accuracy on validation set only achieves 46.67%. Therefore, in this case, the GAGSc method is not effective to identify informative genes for Esophageal cancer data analysis.

Microarray dataset	Number of selected genes	TP	TN	FP	FN	Classification accuracy
Esophageal cancer data	45	7	0	5	3	46.67%

Table 4.14 The classification validation results of GAGSc method on Esophageal cancer data.

Fig. 4.11 shows that the classification performance obtained from 1 round optimization is very low (approximately 50% accurate), while the consistency is too high (nearly 0.5). It seems that these two values are difficult to be improved, because they are only optimized 12 rounds and cannot be effectively improved by GAGSc method.

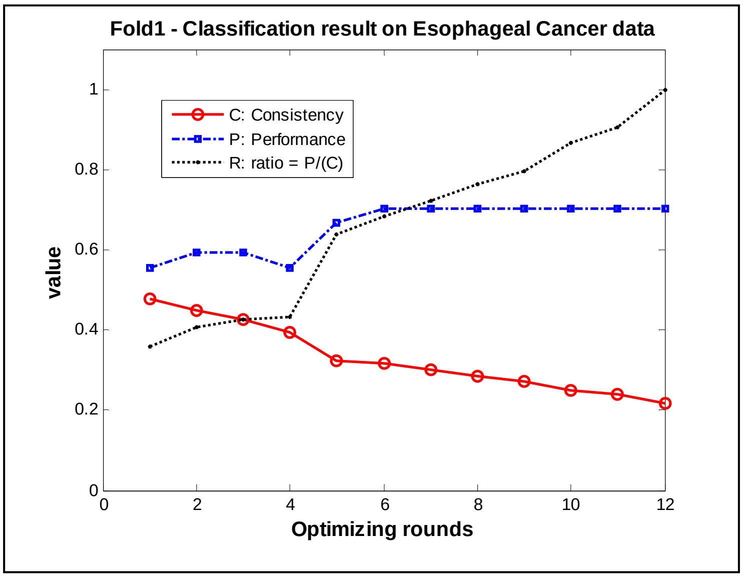


Fig. 4.11 The optimizing results of GAGSc method on Esophageal cancer data.

4.5.1. 9 Classification accuracy summary: GAGSc method vs. publication

For clarity, the classification accuracies obtained by GAGSc method is summarized in Table 4.15, and the reported accuracies in the papers (listed in Table 4.1) is added as well. Our proposed GAGSc method outperforms the published methods on four datasets, and the classification result on colon data is very close to the reported accuracy. However, the classification accuracies of three datasets (CNS, Breast and Esophageal) are significantly lower than the published. As discussed in chapter 1, many published classification results are not based on efficient validation schemes, which results in the experiments are unreplicable and too optimistic. However, the experimental results obtained by the proposed GAGSc method can be easily reproduced, because the totally unbiased validation scheme is applied in this study. These results suggest that reproducible prognosis is possible for only 4 or 5 of the 8 used benchmark datasets.

Data	**Classification accuracy**	
	GAGSc	**Publication**
Lymphoma	**95.84%**	72.5%
Leukaemia	**94.12%**	85%
CNS Tumour	65.00%	83%
Colon Cancer	83.81%	87%
Ovarian	**98.80%**	97%
Breast Cancer	63.16%	94%
Lung Cancer	**91.28%**	90%
Esophageal Cancer	46.67%	93.3%

Table 4.15 Classification accuracy comparison: GAGSc results vs. known results from literature

The proposed GAGSc method described in this chapter have demonstrated that the consistency concept can be used for gene selection to solve the reproducibility problem in microarray data analysis. The main contribution of proposed GAGSc gene selection method is that it ensures the reliability and generalizability of microarray data analysis experiment, and improves the disease classification performance as well. In addition, because GAGSc method does not need previous knowledge about the given microarray data, it can be used as an effective tool in unknown disease diagnosis area.

Here, from the perspective of generalization error, it should be pointed out that the experiments results can be seen as totally unbiased, because the data for validation is independent and never touched in the training process, i.e. before the final informative genes selected, the test data is isolated and has no correlation with these genes.

Therefore, the selected informative genes are entirely fair to any given data for validation. Such a mechanism of gene selection might result in the bad performance in certain microarray datasets, which is due to the characteristic of data. The reported good results in published papers of these datasets are suspect.

4.5.2 LOOLSc method

In this section, to further examine the ability of the proposed consistency concept in gene selection, LOOLSc and LOOLS methods are applied to identify informative genes for classification on Esophageal cancer and lung cancer data.

4.5.2.1 Esophageal cancer data

Table 4.16 ~ 4.17 and Fig. 4.12 show the experimental results obtained by LOOLSc method. Table 4.16 presents the best classification accuracy obtained from the classifier using the informative genes selected by LOOLSc and LOOLS methods. For each gene selection method, the classification accuracies on the independent testing set and training set are presented. The genes selected by LOOLSc method tends to be more effective to express the disease patterns of the Esophageal data than that by LOOLS method, as the classification accuracy on the testing set using the former genes is significantly better than the latter (73.33% vs. 60%). Table 4.17 lists the index number of 15 selected genes used for achieving the best classification accuracy described in table 4.16.

Gene selection method	**LOOLSc** (Testing set / Training set)	**LOOLS** (Testing set / Training set)	Publication
Best classification accuracy	**73.33% /** 88.89%	**60%** / 100%	93.3%

Table 4.16 Summary of the classification results of Esophageal cancer data obtained by LOOLSc and LOOLS. For each gene selection method, the accuracies on independent testing set and on training set are given, *e.g.* the best accuracy on testing set calculated based on 15 genes selected by LOOLSc method is 73.33%, while the accuracy on training set using the same 15 genes is 88.89%.

Methods	**Index of 15 selected genes**														
LOOLSc	157	851	189	384	624	194	94	815	518	445	707	566	291	723	601
LOOLS	432	593	636	442	411	667	40	418	597	144	37	673	427	155	91

Table 4.17 15 informative genes obtained by LOOLSc and LOOLS methods

Fig. 4.12 shows the relationship between the number of genes and the value of consistency and classification performance. The best classification accuracy (73.33%) occurs when 15 genes are selected by LOOLSc method. With LOOLS method, the best accuracy (approximately 65%) and consistency are obtained when 2 genes are selected. An interesting finding is that by using less than 5 genes selected by either LOOLSc or LOOLS, the obtained performance and consistency are very close to the final optimized results, e.g. LOOLSc achieves approximately 65% accuracy by using 4 informative genes. The number of informative genes to be selected will be discussed in chapter 5.

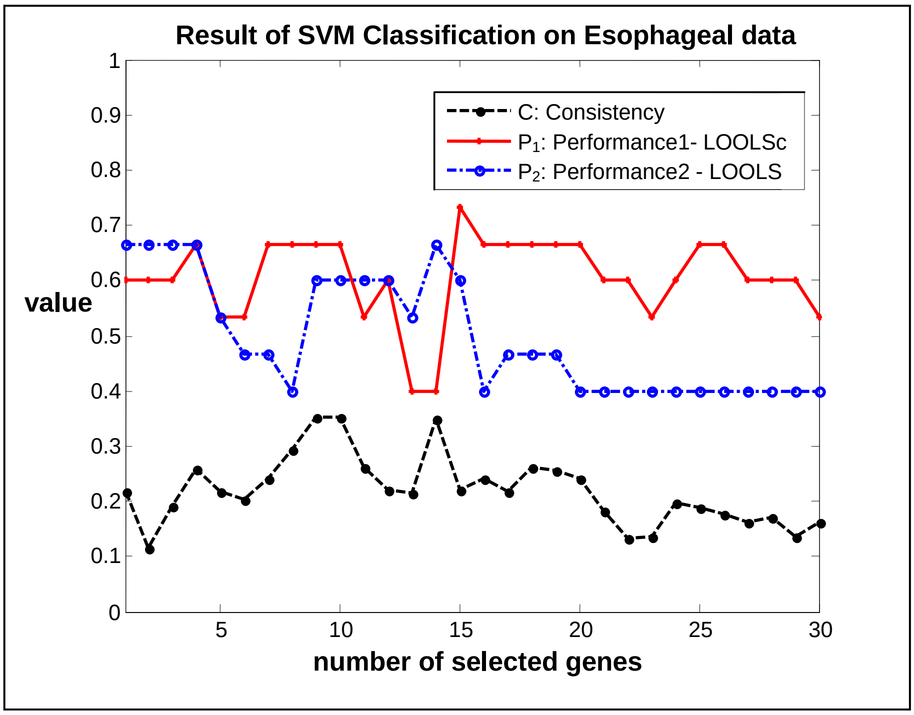


Fig. 4.12 Comparison of the classification results obtained by LOOLSc and LOOLS on Esophageal cancer data. Horizontal axis represents the value of consistency C, classification accuracies P_1 (LOOLSc), P_2 (LOOLS). Vertical axis represents the number of selected genes. Note that C is the consistency calculated in the training process, P_1 and P_2 are the classification accuracies on validating testing set using the informative genes selected by LOOLSc and LOOLS methods, respectively.

4.5.2.2 Lung cancer data

Table 4.18~4.19 and Fig. 4.13 present the experimental results obtained by LOOLSc method on Lung dataset. Table 4.18 shows the best classification accuracy computed using the informative genes selected by LOOLSc and LOOLS gene selection methods. It is clear that LOOLSc method outperforms LOOLS method in terms of classification accuracy (98.66% vs. 89.26%) when 25 informative genes are selected. Table 4.19 lists the index number of 25 selected genes by LOOLSc and LOOLS method.

Gene selection method	**LOOLSc** (Testing set / Training set)	**LOOLS** (Test set / Training set)	Publication
Best classification	**98.66%** / 100%	**89.26%** / 100%	90%

Table 4.18 Summary of the classification results on Lung cancer data obtained by LOOLSc and LOOLS. For each gene selection method, the accuracies on independent testing set and on training set are given, *e.g.* the best accuracy on testing set calculated based on 25 genes selected by LOOLSc method is 98.66%, while the accuracy on training set using the same 25 genes is 100%.

Methods	**Index of 25 selected genes**
LOOLSc	3844, 128, 3, 1, 20, 237, 7, 6, 33, 56, 15, 4, 18, 12, 22, 26, 14, 13, 27, 21, 50, 49, 24, 32, 29
LOOLS	7249, 8213, 6441, 1654, 5540, 7577, 1792, 7570, 11653, 8427, 6537, 9359, 5778, 8381, 8353, 5652, 8592, 11945, 5532, 1429, 11371, 10060, 6991, 4343,

Table 4.19 25 informative genes obtained by LOOLSc and LOOLS methods

Fig. 4.13 shows that the classification accuracies using the informative genes selected by LOOLSc method are higher than that by LOOLS method. The best classification accuracy is produced by the classifier with 25 informative genes selected by LOOLSc method (98.66%). Additionally, Lund cancer data has a stable and good consistency characteristic, which ensures the high classification accuracies obtained on the independent testing sets with different number of selected genes.

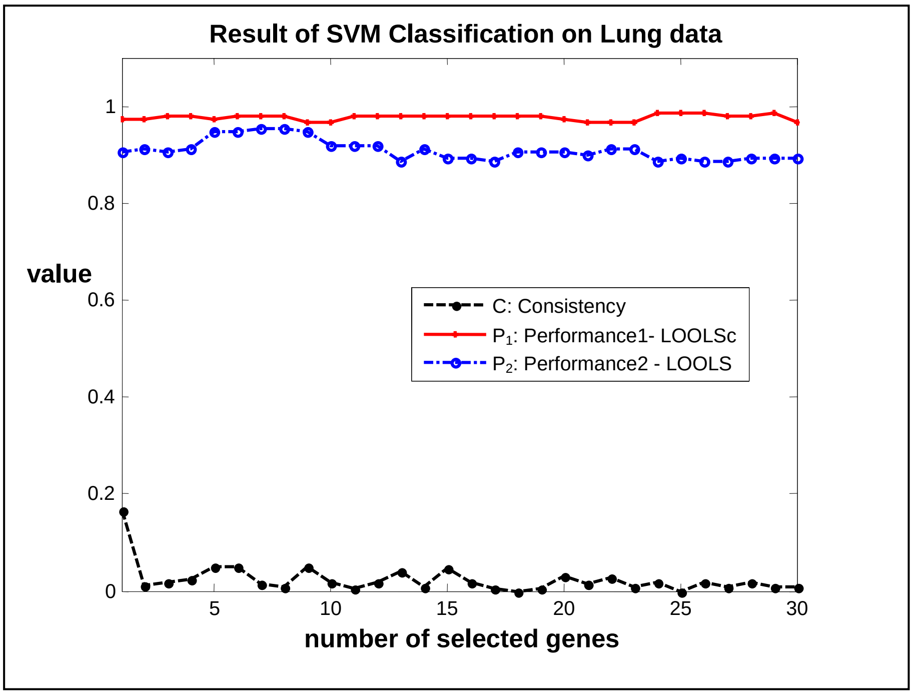


Fig. 4.13 Comparison of the classification results obtained by LOOLSc and LOOLS on Lung cancer data.

4.5.2.3 Summary of LOOLSc and LOOLS methods

This experiment shows that the LOOLSc method outperforms LOOLS method in terms of classification accuracy on both Esophageal cancer and Lung cancer data. It turns out that the utilization of consistency concept in gene selection benefits the unbiased microarray data analysis. With LOOLSc method, the better classification accuracy occurs when the consistency value is close to zero (the best consistency). Unlike many disputed good experiment results reported in microarray analysis literature, all the experimental results of LOOLSc method described in this thesis are reproducible.

Most importantly, the results demonstrates that a good classification performance on the training set does not give the same performance on the testing set, which follows that the consistency is very essential on achieving an unbiased and reproducible microarray data analysis result.

Chapter 5
Discussion and Conclusions

5.1 Discussion

5.1.1 The number of biomarker genes

The number of biomarker genes to be selected in GAGSc is a big concern, because it determines the GA searching procedure and the final reliability and effectiveness of gene selection. The issue raises a question: How many genes should be selected in the microarray data analysis? As there is no agreement on how many informative genes are able to differentially express disease patterns (Mukherjee, Roberts, & Lann, 2005) so far, it is assumed that the best number of selected genes in GAGSc should satisfy the following criterion based on the results of previous experiments:

For a given dataset *D* pertaining to a classification task, the number of genes is best fit to *D*, when the consistency and performance are balanced at one point. Fig. 5.1 schematically illustrates the condition of best number of selected genes, where *β* is the balanced point between consistency and diagnosis prediction accuracy.

Theoretically, the balanced point *β* can be reached by a gene selection function $f_s(F_{sc}, F_{sp})$, when the following criteria are satisfied:

$$\lim_{t \to n} \left(F_{sc}(s_t, t) - F_{sc}(s_{t+1}, t+1) \right) \to \delta_1 \tag{5.1}$$

$$\lim_{t \to n} \left(F_{sp}(s_t, t) - F_{sp}(s_{t+1}, t+1) \right) \to \delta_2 \tag{5.2}$$

where:

1. F_{sc}, F_{sp} are the functions for computing consistency and diagnosis accuracy, under the condition of gene selection, respectively.
2. δ_1 and δ_2 are two pre-specified thresholds for evaluating the improvement of consistency and diagnosis accuracy.
3. *t* is the number of candidate genes to be selected.
4. *n* is the number of genes in the given microarray dataset.
5. s_t, s_{t+1} are the sets of *t* and *t+1* selected candidate genes, respectively.

As shown in Fig. 5.1, it is a multi-optimization problem to determine the number of biomarker genes. In this study, prediction accuracy and consistency are equally important, and a ratio is introduced for optimizing them simultaneously. However, prediction accuracy and consistency may have different priorities in the optimization process for gene selection, in terms of different end users. For example, clinical end-users are probably more concentrated on prediction accuracy, while bioinformatics researchers are more interested in consistent gene selection modeling.

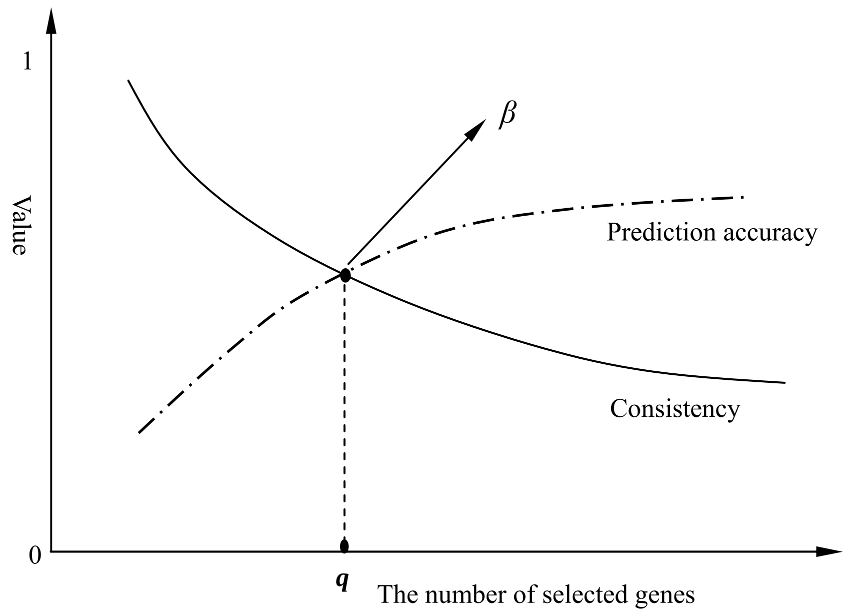


Fig. 5.1 The expected relationship among consistency, diagnosis accuracy, and the number of selected genes. The value of diagnosis accuracy and consistency are shown vs. the number of selected genes. Point β represents the condition of the best number of genes occurs when the curves of performance and consistency are intersected. The number $\boldsymbol{q}$ is the best number of genes to be selected for classification on this given microarray dataset.

A reliable set of informative genes can contribute to a better understanding of a microarray dataset pertaining to a biological task. In practice, the robustness and reliability of selected informative genes are often limited by the microarray datasets with the very small number of samples. For example, there are only 27 samples in Esophageal cancer data, which is very little information to identify informative genes for constructing classifier. Ein-Dor et al. (2006) argued that thousands of samples are required to find a reliable and robust gene set for microarray data analysis in cancer diagnosis using conventional analysis methods. However, most real microarray datasets have only tens of samples, which is far from that required. Thousands of samples are unrealistic for real microarray experiments. Instead, more efficient resampling methods should be applied to generate more sets for training.

5.1.2 Initial gene set in GAGSc method

One of the major challenges of GAGSc is how many genes should be selected as an initial gene set of GA optimization. As discussed in section 5.1, the efficiency of GAGSc method is in relation to a proper number of initial candidate genes, and these initial genes can reduce the GA searching time in gene selection.

In GAGSc method, all genes of a given dataset are initially divided into several segments (see Fig. 3.6), and for each segment, one gene is set as the candidate gene of GAGSc. Thus, a proper number of segments determine the efficiency of gene selection process. If there are too many initially divided segments, GAGSc will spend longer time searching candidate gene and evaluating the importance of each candidate gene. In contrast, if the number of initial segments is too small, gene space cannot be searched by GAGSc due to less genes are able to be reached by the evolutionary gene selection procedure.

According to the guidelines introduced by Jain et al. (2000) that when researchers aims to build an exact relationship between the probability of misclassification, the number of training samples and the number of genes, a ratio regarding to the sample size to dimensionality should be considered. Generally, the following ratio is recommended for microarray data analysis:

$$(n/k)/m > 5 \tag{4.1}$$

where *n* is the number of samples, *k* is the number of classes and *m* is the number of genes. For example, if there is a two-class microarray dataset with 100 samples for classification problem, not more than 10 important genes should be found to construct the classifier with acceptable generalizability performance. However, this ratio is often practically violated in reported research, because the number of samples in many microarray datasets is less than 100, and the best classification is thought generally to occur when 15 to 40 genes are selected (Li, & Yang, 2002).

As GAGSc is a self-optimizing process, the number of finally selected genes will be adjusted in the evolution procedure. The initial number of genes in this thesis is set to 20, because 15 to 30 genes are usually used in gene selection in literature (Li et al., 2002; Tang, Suganthan, & Yao, 2006).

5.1.3 Common genes experiment

The experiment of common genes is proposed to determine the number of initial gene set. The experiment is motivated by the concept of common genes appearing in two gene selection tests.

The common genes experiment is applied on three datasets, esophageal, colon, and leukaemia. *m* is assigned as the initial number of genes to be selected by GAGSc method. The final selected informative genes are denoted as *S*. The common genes experiment can be summarized into the following steps:

(a) GAGSc method starts to select *m* genes for further GA optimization. The final selected informative genes by GAGSc method are grouped into *S1*.

(b) GAGSc repeat step(a) gene selection, and obtain another set of selected informative genes, *S2*.

(c) The common genes are the genes in common between *S1* and *S2*, which are defined as *CmSet*.

(d) Increase the initial number of genes for evaluating by GAGSc ($m = m + j$), where j is set to 20.

(e) Repeat steps (a) – (d) *i* times, and return *i* groups of common genes in terms of different number of selected informative genes.

For clarity, a flowchart is given in Fig. 5.2.

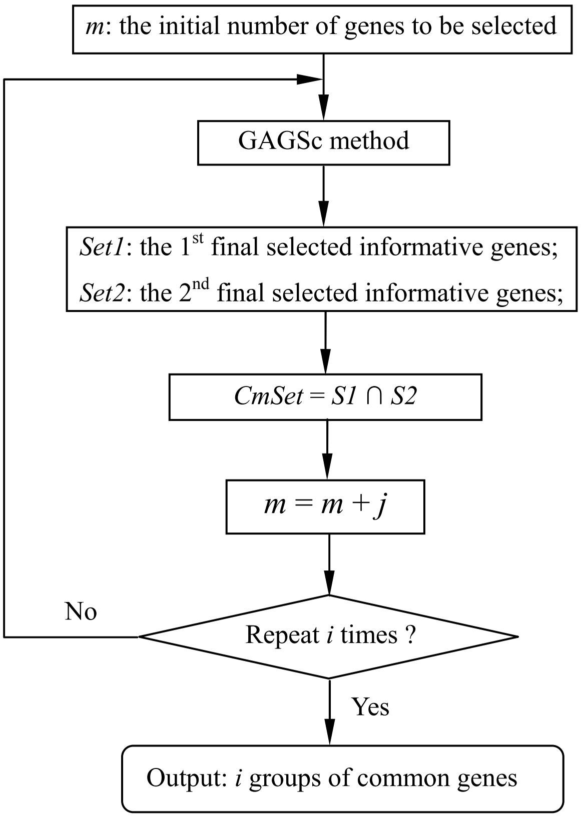

m: the initial number of genes to be selected
GAGSc method
Set1: the 1st final selected informative genes;
Set2: the 2nd final selected informative genes;
CmSet = *S1* ∩ *S2*
$m = m + j$
No
Repeat *i* times ?
Yes
Output: *i* groups of common genes

Fig. 5.2. The simple flowchart of common genes experiment

The tests on Esophageal cancer, colon cancer and leukaemia datasets are recorded in Table 5.1. Generally, the more genes selected, the more genes in common are found in two selected gene sets. However, as compared with the number of finally selected genes, the amount of common genes is obviously small. For example, for Leukaemia data, when nearly 300 genes are selected from two tests, only 59 genes (approximately 20% of final selected genes) are found in common between sets. Even in the most optimistic case, 183 out of approximately 300 genes are found in common in the two tests on Esophageal data. This approach of evaluating common genes doesn't perform well in this test. A more efficient method for determining the initial gene set is thus necessary, and is expected to be developed in future work.

Data	Common genes	Selected informative genes		The initial number of genes to be selected (m)
		Set1	*Set2*	
Esophageal cancer data	7	50	41	10
	28	94	86	30
	44	133	128	50
	89	222	225	70
	183	327	314	90
Colon data	6	24	14	10
	7	33	49	30
	12	61	69	50
	24	100	103	70
	40	149	143	90
Leukaemia data	2	15	30	10
	11	55	64	30
	19	106	117	50
	37	191	200	70
	59	301	292	90

Table 5.1 The results summary of common genes experiment

5.1.4 Limitation

The main two limitations of proposed gene selection methods are from two perspectives: computational complexity and generalizability. The extremely computational complexity makes of LOOLSc method is difficult to be applied on huge dimensionality microarray data (more than 10,000 genes) in practice.

5.1.4.1 Computational complexity

The experimental results have shown that the informative genes selected by GAGSc and LOOLSc methods can significantly improve the classification accuracy and reproducibility for microarray data analysis. Nevertheless, GAGSc and LOOLSc methods are found very costly in terms of computational complexity in the experiments. For GAGSc method, it usually takes 12~16 hours to get the final selected genes on a computer with 3.2 GHz P4 CPU and 2048 MB RAM, when the microarray dataset contains more than 7,000 genes. This is mainly due to the evolutionary function, including mutation and crossover operation in candidate genes selection procedure.

The experiment of LOOLSc was much more expensive with respect to the time cost. Two datasets were used in the evaluation experiments of LOOLSc, because it took about several hours to measure the effectiveness of one candidate gene in SVM classification stage depending on the number of genes in the given dataset. For example, it takes LOOLSc method approximately 1 week to select 20 informative genes out of 12,533 genes in Lung cancer data analysis. However, for relatively small dimensional microarray data, such as Esophageal cancer data (with 859 genes), the computation complexity is decreased sharply (only 10 hours to get 30 selected genes).

5.1.4.2 Generalizability

The totally unbiased verification scheme has been used for all experiments in this thesis to avoid the bias error. Therefore, all experimental results obtained through the proposed GAGSc and LOOLSc methods are repeatable. However, it is found that GAGSc method is not robust as expected on some microarray datasets, such as Esophageal cancer, CNS tumour and Breast cancer datasets. LOOLSc method has been only examined over two cancer data analysis, so that its generalizability needs further study.

Note that this issue is mainly caused by the lack of sufficient information from the microarray data during the classifier training procedure. To achieve an accurate prediction outcome with good reproducibility, using a small number of samples cannot identify a reliable and effective gene list (Ein-Dor, Zuk, & Domany, 2006; Michiels, Koscielny, & Hill, 2005).

5.1.4.3 Gene selection performance measuring method

Another limitation is that only classification accuracy is used as the measurement for evaluating gene selection performance. This measuring method is capable of estimating the misclassification costs of diagnosis, but cannot efficiently evaluate the bias error occurring in the experiments. In this thesis, the classification accuracy is used to measure the capability of gene selection because of the following two reasons:

1). There have been no mature and effective performance metrics applicable for gene selection methods so far (Statnikov, Aliferis, Tsamardinos, Hardin et al., 2005), and accuracy has been widely used in the published studies.

2). Using accuracy can simplify the statistical comparison results and is easy to interpret.

However, more effective measurement scheme for gene selection method should be considered, which can measure the classification performance as well as the bias error.

5.2 Conclusions

In conclusion, the overall objectives of this study are to (1) investigate the proposed performance-based consistency concept, (2) develop new gene selection methods (GAGSc and LOOLSc) in the proposed performance-based consistency theory. The findings of experiments have demonstrated that the utilization of proposed consistency concept substantially improves the classification accuracy of microarray data analysis for cancer diagnosis.

One contribution of the proposed consistency concept is that it can be easily incorporated into more sophisticated gene selection systems to enhance the overall performance of microarray data analysis. For example, the consistency concept can be implemented to the gene selection methods based on either Neuro-computing or traditional statistical algorithms to identify more reliable and robust informative genes.

The second contribution is using the proposed gene selection methods (GAGSc and LOOLSc), the final selected informative genes are capable of constructing better classifier for disease diagnosis in terms of prediction accuracies. The unbiased prediction accuracies on eight benchmark datasets obtained by GAGSc method in this study are very competitive to the reported results in literature. Note that some of the published prediction results are not validated on independent datasets, and thus remain suspect.

Finally, the utilization of proposed consistency concept can benefit the good reproducibility of microarray experiments. In this thesis, the proposed GAGSc method and LOOLSc methods are developed on the basis of totally unbiased validation schemes, which ensure the achieved good experimental results are reproducible.

5.3 Future work

The findings of this study indicate that the proposed consistency concept in gene selection is a useful innovation in several areas. Considerations for further improving the gene selection methods based on the proposed consistency concept are:

(1) Incorporate clustering in the pre-process stage of gene selection

In this study, the huge computational complexity is one of the main limitations of the proposed GAGSc method. The problem is manly due to the massive number of genes, which makes the GA search is extremely time costly. The cluster algorithms can be used before the GA search to find a certain number of clusters. Then these clusters can be used for determining either the initial number of genes to be selected or the number of finally selected informative genes. This is expected to improve the efficiency of GA search engine, so that the computational complexity can be significantly reduced.

(2) Employ different data sampling methods

In this study, K-fold cross-validation is used for estimating generalization error for gene selection. As discussed in chapter 3, bootstrap is another popular sampling method for small sample size dataset. In the future work, bootstrap sampling techniques are expected to improve the reproducibility of microarray data analysis.

References:

Alizadeh, A. A., Eisen, M., et al. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature, 403*(6769), 503-511.

Allison, D., Cui, X., et al. (2006). Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews Genetics, 7*(1), 55-65.

Alon, U., Barkai, N., et al. (1999). *Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays.* Paper presented at the Proc Natl Acad Sci, USA.

Ambroise, C., & McLachlan, G. J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *PNAS, 99*(10), 6562-6566.

Arfin, S., Long, A. D., et al. (2000). Global Gene Expression Profiling in Escherichia coli K12: The effects of integration host factor. *J. Biol. Chem., 275*(38), 29672-29684.

Asyali, M. H., Colak, D., et al. (2006). Gene Expression Profile Classification: A Review. *Current Bioinformatics 1*, 55-73.

Atler, O., & et al. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *P.N.A.S., 97*(18), 10101-10106.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Statist. Soc., B(57), 289–300.

Ben-Dor, A., Friedman, N., et al. (2001). Class discovery in gene expression data. *RECOMB*, 31-38.

Binder, S. R., Genovese, M. C., Merrill, J. T., et al. (2005). Computer-Assisted Pattern Recognition of Autoantibody Results. Clin Diagn Lab Immunol, 12(12), 1353-1357.

Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze, 8*, 3-62.

Braga-Neto, U., Hashimoto, R., et al. (2004). Is cross-validation better than resubstitution for ranking genes? *Bioinformatics, 20*(2), 253-258.

Breiman, L., & Spector, P. (1992). Submodel selection and evaluation in regression: The X-random case60. *International Statistical Review, 60*, 291-319.

Chuang, H.-Y., Liu, H., et al. (2004). *Identifying Significant Genes from Microarray Data. BIBE 2004: 358-365.* Paper presented at the BIBE 2004. Proceedings. Fourth IEEE Symposium.

Crimins, F., Dimitri, R., Klein, T., et al. (2002). Higher-Dimensional Approach for Classification of Lung Cancer Microarray Data. Paper presented at the Classification Society of North America (CSNA), 2002 Annual Conference.

Cui, B., Ooi, B. C., Su, J., et al. (2003). Contorting high dimensional data for efficient main memory KNN processing. Paper presented at the 2003 ACM SIGMOD international conference on Management of data, San Diego, California.

Devijver, P. A., & Kittler, J. (1982). *Pattern recognition: a statistical approach*. London: Prentice-Hall Inc.

Ding, C. (2002). *Analysis of gene expression profiles: class discovery and leaf ordering.* Paper presented at the Annual Conference on Research in Computational Molecular Biology Proceedings of the sixth annual international conference on Computational biology, Washington, DC, USA.

Ding, C., & Peng, H. (2003). *Minimum Redundancy Feature Selection for Gene Expression Data.* Paper presented at the Proc. IEEE Computer Society Bioinformatics Conference (CSB 2003), Stanford, CA.

Draghici, S., Kulaeva, O., et al. (2003). Noise sampling method: an ANOVA approach allowing robust selection of differentially regulated genes measured by DNA microarrays. *Bioinformatics, 19*(11), 1348-1359.

Dudoit, S., Fridlyand, J., et al. (2000). *Comparison of discrimination methods for the classification of tumors using gene expression data*: UC Berkeley.

Dudoit, S., Yang, Y. H., et al. (2002). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Stat. Sinica, 12,* 111-139.

Efron, B. (1979). Bootstrap methods: another look at the jacknife. *Annals of Statistics, 7*(1), 1-26.

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *J. of the American Statistical Association, 78,* 316-331.

Efron, B., Tibshirani, R., et al. (2001). Empirical Bayes Analysis of a Microarray Experiment. *Journal of the American Statistical Association, 96,* 1151-1160.

Ein-Dor, L., Zuk, O., et al. (2006). Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *PNAS, 103*(15), 5923-5928.

Eisen, M. B., Spellman, P. T., et al. (1998). *Cluster analysis and display of genome-wide expression patterns.* Paper presented at the Proc Natl Acad Sci USA.

Fawcett, T. (2003). *ROC Graphs: Notes and Practical Consideration for Researchers* (Technical report No. HPL2003 -4): HP Laboratories.

Fix, E., & Hodges, J. L. (1951). *Discriminatory analysis---nonparametric discrimination: Consistency properties* (No. 4). Randolph Field, Texas: USAF School of Aviation Medicine.

Fung, G. M., & Mangasarian, O. L. (2004). A Feature Selection Newton Method for Support Vector Machine Classification. *Computational Optimization and Applications, 28*(2), 185-202.

Furey, T., Cristianini, N., et al. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics, 16*(10), 906-914.

Galvin, J. E., & Ginsberg , S. D. (2004). Expression profiling and pharmacotherapeutic development in the central nervous system. *Alzheimer Dis. Assoc. Disord., 18,* 264-269.

Galvin, J. E., Powlishta, K., et al. (2005). Predictors of preclinical Alzheimer disease and dementia: a clinicopathologic study. *Arch Neurol, 62*(5), 758-765.

Glymour, C., Madigan, D., et al. (1996). Statistical Inference and Data Mining. *Communication of the ACM, 39*(11), 35-41.

Goh, L., Song, Q., et al. (2004). *A novel feature selection method to improve classification of gene expression data.* Paper presented at the ACM International Conference Proceeding Series: Proceedings of the second conference on Asia-Pacific bioinformatics - Volume 29, Dunedin, New Zealand.

Goldberg, D. E. (1989). *GeneticAlgorithm in Search, Optim, zation and Machine Learning*. MA: Addison-Wesley.

Golub, T. R., Slonim, D. K., et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science, 286,* 531-537.

Gordon, G. J., Jensen, R., et al. (2002). Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gege Expression Ratios in Lung Cancer And Mesothelioma. *Cancer Research, 62,* 4963-4967.

Guan, Z., & Zhao, H. (2005). A semiparametric approach for marker gene selection based on gene expression data. *Bioinformatics, 21*(4), 529-536.

Gunn, S. (1997). *Support Vector Machines for Classification and Regression*: Image Speech and Intelligent Systems Research Group, University of Southampton.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research, 3,* 1157-1182.

Guyon, I., Weston, J., et al. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning, 46*(1), 389-422.

Hamamoto, Y., Uchimura, S., et al. (1996). On the behavior of artificial neural network classifiers in high-dimensional spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(5), 571-574.

Hastie, T., & et al. (2000). 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology, 1*(2), 1-21.

Hayashida, Y., Honda, K., et al. (2005). Possible Prediction of Chemoradiosensitivity of Esophageal Cancer by Serum Protein Profiling. *Clin Cancer Res, 11*(22), 8042-8047.

Hero, A. (2003). *Gene selection and ranking with microarray data.* Paper presented at the Proc. of Intl Conf on Signal Processing and Applications, Paris.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*: The University of Michigan Press

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scand. J. Statist., 6,* 65-70.

Hopfield, J., & Tank, D. W. (1985). "Neural" computation of decisions in optimization problems. *Biological Cybernetics, 52*(3), 141-152.

Hosking, J., Pednault, E., et al. (1997). A statistical perspective on data mining. *Future Generation Computing System, 13*(2), 117-134.

Huerta, E. B., Duval, B., et al. (2006). A hybrid GA/SVM approach for gene selection and classification of Microarry data. *Lecture Notes in Computer Science, 3907,* 34-44.

Inza, I., Larranaga, P., et al. (2004). Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial Intelligence, 31*(2), 91-103.

Ioannidis, J. P. A. (2005). Microarrays and molecular research: noise discovery? *Lancet, 365,* 453-455.

Ioannidis, J. P. A., Trikalinos, T. A., et al. (2003). Genetic associations in large versus small studies: an empirical assessment. *The Lancet, 361*(9357), 567-571.

Iwao-Koizumi, K., Matoba, R., et al. (2005). Prediction of Docetaxel Response in Human Breast Cancer by Gene Expression Profiling. *American Society of Clinical Oncology 33*(3), 422-431.

Jaeger, J., Sengupta, R., et al. (2003). *Improved gene selection for classification of microarrays.* Paper presented at the Pacific Symposium on Biocomputing, Kauai, Hawaii.

Jain, A. K., Duin, R., et al. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Patt. Anal. and Machine Intell, Patt. Anal. and Machine Intell.,* *Vol. 22, No. 1, pp. 4-37, 2000*(1), 4-37.

Kasabov, N., Middlemiss, M., et al. (2003). *A generic connectionist-based method for on-line feature selection and modelling with a case study of gene expression data analysis.* Paper presented at the Conferences in Research and Practice in Information Technology Series: Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics 2003 - Volume 19, Adelaide, Australia.

KEDRI. (2002). NeuCom. from www.theneucom.com

Kim, K.-Y., Kim, B.-J., & Yi, G.-S. (2004). Reuse of imputed data in microarray analysis increases imputation efficiency. BMC Bioinformatics, 5(160).

Kohavi, R. (1995). *A study of cross-validation and bootstrap for accuracy estimation and model selection.* Paper presented at the International Joint Conference on Artificial Intelligence (IJCAI), Montreal, Quebec, Canada.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*(1-2), 273-324.

Lai, C., Reinders, M., et al. (2004). *On univariate selection methods in gene expression datasets.* Paper presented at the Tenth Annual Conference of the Advanced School for Computing and Imaging, Port Zelande, The Netherlands.

Lee, K. E., Sha, N., et al. (2003). Gene selection: a Bayesian variable selection approach. *Bioinformatics, 19*(1), 90-97.

Levene, H. (1960). Robust Tests for Equality of Variances. In I. Olkin & P. Alto (Eds.), *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling* (pp. 278-292). Stanford, CA: Stanford University Press.

Li, L., Weinberg, C. R., et al. (2001). Gene Selection for sample calssification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics, 17*(12), 1131-1142.

Li, W., & Xiong, M. (2002). Tclass: tumor classification system based on gene expression profile. *Bioinformatics, 18*(2), 325-326.

Li, W., & Yang, Y. (2002). How many genes are needed for a discriminant microarray data analysis? In S. M. Lin & K. F. Johnson (Eds.), *Methods of Microarray Data Analysis* (pp. 137-150): Kluwer Academic.

Liotta, L. A., Lowenthal, M., et al. (2005). Importance of Communication Between Producers and Consumers of Publicly Available Experimental Data. *J Natl Cancer Inst 2005, 97*, 310-314.

Louis, S. J. (1993). *Genetic Algorithms as a Computational Tool for Design.* Indiana University.

Mandel, S., Weinreb, O., et al. (2003). Using cDNA microarray to assess Parkinson's disease models and the effects of neuroprotective drugs. *Trends Pharmacol Sci., 24*(4), 184-191.

Mangasarian, O. L., & Musicant, D. R. (2001). Lagrangian support vector machines. *The Journal of Machine Learning Research, 1*, 161-177.

Michiels, S., Koscielny, S., et al. (2005). Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet, 365*, 488-492.

Mitchell, M., & Forrest, S. (1994). Genetic algorithms and artificial life. *Artificial Life, 1*(3), 267-289.

Model, F., Adorján, P., et al. (2001). Feature selection for DNA methylation based cancer classification. *Bioinformatics, 17*, S157-S164.

Mukherjee, S. (2003). *Classifying Microarray Data Using Support Vector Machines*. Heidelberg: Springer-Verlag.

Mukherjee, S., & Roberts, S. J. (2004). *Probabilistic Consistency Analysis for Gene Selection.* Paper presented at the CSB, Stanford, CA, USA.

Mukherjee, S., Roberts, S. J., et al. (2005). Data-adaptive test statistics for microarray data. *Bioinformatics, 00*(00), 1-7.

Pang, S., Kim, D., et al. (2005). Face membership authentication using SVM classification tree generated by membership-based LLE data partition. *IEEE Trans Neural Neural Network, 16*(2), 436-446.

Pawitan, Y., Murthy, K. R. K., et al. (2005). Bias in the estimation of false discovery rate in microarray studies. *Bioinformatics, 21*(20), 3865-3872.

Petricoin, E. F., Ardekani, A. M., et al. (2002). Use of proteomic patterns in serum to identify ovarian cancer. *Lancet, 359*, 572-577.

Pomeroy, S., Tamayo, P., et al. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature, 415*(6870), 436-442.

Qiu, X., Xiao, Y., et al. (2006). Assessing Stability of Gene Selection in Microarray Data Analysis. *BMC Bioinformatics, 7*(50).

Ramaswamy, S., & Perou, C. (2003). DNA microarrays in breast cancer: the promise of personalised medicine. Lancet, 361(9369), 1590-1596.

Ransohoff, D. F. (2004). Rules of evidence for cancer molecular marker discovery and validation. *Nature Reviews Cancer, 4*, 309-314.

Ransohoff, D. F. (2005a). Bias as a threat to the validity of cancer molecular-marker research. *Nat Rev Cancer, 5*(2), 142-149.

Ransohoff, D. F. (2005b). Lessons from controversy: Ovarian cancer screening and serum proteomics. *Journal of National Cancer Institute, 97*(4), 315-319.

Reiner, A., Yekutieli, D., et al. (2003). Identifying differentially expressed genes using false discovery rate controlling procedures. *Bioinformatics, 19*(3), 368-375.

Schena, M. (2002). *Microarray analysis*. New York: John Wiley & Sons.

Shaffer., J. P. (1995). Multiple hypothesis testing. *Annu. Rev. Psychol., 46*, 561-584.

Shaffler, J. P. (1986). Modifed sequentially rejective multiple test procedures. *JASA, 81*, 826-831.

Snedecor, G. W., & Cochran, W. G. (1989). *Statistical Methods* (Eighth ed.): Iowa State University Press.

Statnikov, A., Aliferis, C. F., et al. (2005). A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics, 21*(5), 631-643.

Sun, Z., Bebis, G., et al. (2002). *Genetic feature subset selection for gender classification: a comparison study.* Paper presented at the IEEE Workshop on Applications of Computer Vision.

Suykens, J. A. K., & Vandewalle, J. (1999). Least Squares Support Vector Machine Classifiers. *Neural Processing Letters, 9*(3), 293-300.

Tamayo, P., & et al. (1999). Interpreting patterns of gene expression with self-organizing maps. *P.N.A.S., 96*(6), 2907-2912.

Tanaka, T. S., Jaradat, S. A., et al. (2000). *Genome-wide expression profiling of mid-gestation placenta and embryo using a 15,000 mouse developmental cDNA microarray.* Paper presented at the Proc. Natl. Acad. Sci.

Tang, E. K., Suganthan, P., et al. (2006). Gene selection algorithms for microarray data based on least squares support vector machine. *BMC Bioinformatics, 7*(95).

Thomas, J. G., Olson, J. M., et al. (2001). An Efficient and Robust Statistical Modeling Approach to Discover Differentially Expressed Genes Using Genomic Expression Profiles. *Genome Research, 11*(7), 1227-1236.

Tibshirani, R. J. (2006). A simple method for assessing sample sizes in microarray experiments. *BMC Bioinformatics, 7*(106).

Triola, M. (1998). *Elementary Statistics* (Seventh ed.): Addison Wesley Longman, Inc.

Tusher, V., Tibshirani, R., et al. (2001). Significance analysis of microarrays applied to the ionizing radiation response.

van't Veer, L., Dai, H., et al. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature, 415*(6871), 530-536.

Vapnik, V. (1998). *Statistical Learning Theory*: Wiley-Interscience, NY, USA.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics, 7*(91).

Veer, L. J. v. t., Dai, H., et al. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature, 415*, 530-536.

Wagner. (2004). A test before its time? FDA stalls distribution process of proteomic test. *J Natl Cancer Inst., 96*(7), 500-501.

Weinberger, K. Q., Blitzer, J., & Saul, L. K. (2005). Distance Metric Learning for Large Margin Nearest Neighbor Classification. Paper presented at the Neural Information Processing Systems, Vancouver, Canada.

Welch, B. L. (1938). The significance of the difference between two means when the population variances are unequal. *Biometrika, 29*, 350-362.

Westfall, P., & Young, S. (1993). *Resampling-based multiple testing: examples and methods for multiple P-value adjustment*. New York: John Wiley, Sons.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics, 1*(80-83).

Wolf, L., Shashua, A., et al. (2004). *Selecting relevant genes with a spectral approach* (No. CBCL Paper No.238). Cambridge, MA, USA: Massachusetts Institute of Technology.

Wu, B. (2005). Differential gene expression detection using penalized linear regression models: the improved SAM statistics *Bioinformatics, 21*(8), 1565-1571.

Xia, C., Hsu, W., et al. (2005). *Paper session IR-6 (information retrieval): IR models 1: ERkNN: efficient reverse k-nearest neighbors retrieval with local kNN-distance estimation.* Paper presented at the 14th ACM international conference on Information and knowledge management CIKM '05, Bremen, Germany.

Zhao, Y., & Kwoh, C. K. (2006). Fast leave-one-out evaluation for dynamic gene selection. *Soft Comput, 10*, 346-350.

Zhou, X., & Mao, K. Z. (2005). LS Bound based gene selection for DNA microarray data. *Bioinformatics, 21*(8), 1559-1564.

Zhu, W., Wang, X., et al. (2003). Detection of cancer-specific markers amid massive mass spectral data. *PNAS, 100*, 14666-14671.