Markov Blanket-Embedded Genetic Algorithm for Gene Selection

Zexuan Zhu [a,b], Yew-Soon Ong [a,*], Manoranjan Dash [a]

[a] *Division of Information Systems, School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798*

[b] *Bioinformatics Research Centre, Nanyang Technological University, Research TechnoPlaza, 50 Nanyang Drive, Singapore 637553*

Abstract

Microarray technologies enable quantitative simultaneous monitoring of expression levels for thousands of genes under various experimental conditions. This new technology has provided a new way of biological classification on a genome-wide scale. However, predictive accuracy is affected by the presence of thousands of genes many of which are unnecessary from the classification point of view. So, a key issue of microarray data classification is to identify the smallest possible set of genes that can achieve good predictive accuracy. In this study, we propose a novel Markov blanket-embedded genetic algorithm (MBEGA) for gene selection problem. In particular, the embedded Markov blanket based memetic operators add or delete features (or genes) from a genetic algorithm (GA) solution so as to quickly improve the solution and fine-tune the search. Empirical results on synthetic and microarray benchmark datasets suggest that MBEGA is effective and efficient in eliminating irrelevant and redundant features based on both Markov blanket and predictive power in classifier model. We take representative methods from each of filter, wrapper, and standard GA and show that MBEGA gives better overall performance than the existing counterparts in terms of all four evaluation criteria, i.e., classification accuracy, number of selected genes, computational cost, and robustness.

Key words: Microarray, Feature Selection, Markov Blanket, Genetic Algorithm (GA), Memetic Algorithm (MA)
PACS:

* Corresponding author. Tel.: +65 67906448; fax: +65 67926559
Email address: asysong@ntu.edu.sg (Yew-Soon Ong).

1 Introduction

Microarray technology has attracted increasing interest in many academic communities and industries over the recent years. This breakthrough in technology promises a new insight into the mechanisms of living systems by providing a way to simultaneously measure the activities and interaction of thousands of genes. For example, obtaining genome-wide expression data from cancerous tissues provides clues for cancer classification and accurate diagnostic tests. Machine learning techniques have been successfully applied to cancer classification using microarray data [1–21]. A significant amount of new discoveries have been made and new biomarkers for various cancer have been detected from the microarray data analysis. However, cancer classification has remained a great challenge to computer scientists. The main difficulties lie in the nature of the microarray gene expression data, which is inherently noisy and high-dimensional. Natural biological fluctuations are likely to import measurement variations and bring implications to microarray analysis. In addition, the microarray experiment involves complex scientific procedures during which errors are commonly introduced due to the imperfections of instruments, impurity of materials or negligence of scientist. Microarray data is also high-dimensional with thousands of genes but with only small number of instances [1] available for analysis. This makes learning from microarray data an arduous task under the effect of curse of dimensionality. Furthermore, microarray data often contains many irrelevant and redundant features, which affect the speed and accuracy of most learning algorithms.

Feature selection, also known as gene selection in the context of microarray data analysis, addresses the aforementioned problems by removing the irrelevant and redundant features. This helps improve the prediction performance of the trained model, reduce the computational cost requirement and at the same time provide a better understanding of the data [22–24]. Generally, a typical feature selection method consists of four components: a subset generation or search procedure, an evaluation function, a stopping criterion, and a validation procedure. This general process of feature selection is illustrated in Figure 1. Depending on whether an inductive algorithm is used for feature subset evaluation, feature selection algorithms are widely categorized into two groups: filter and wrapper methods [25].

Traditional gene selection often uses filter methods, which is independent with the induction algorithm. They rank genes according to their individual relevance or discriminative power with respect to the target classes and thus, they select top ranked genes [2,5]. Wrapper methods, on the contrary, use the induction algorithm itself to evaluate the candidate feature subsets. They

[1] An instance here means a sample test in the microarray terminology.

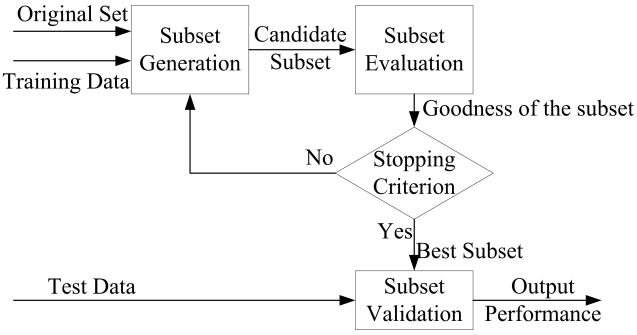

Fig. 1. General procedure of feature selection

generally select feature subsets more suitable for the induction algorithm than the filter methods.

A key issue for feature selection algorithm is how to search the space of feature subsets which is exponential in the number of features. For a survey on the different search methods (e.g., complete search, heuristic search, and random search) used in feature selection, the reader is referred to [22,24]. These methods have shown promising results in a number of real world applications. However, on microarray data, as the number of genes (features) are typically very large, most of these existing methods face the problems of intractable computational time.

Genetic Algorithm (GA)[26], one of the commonly used modern stochastic global search technique, has well known ability to produce high quality solution within tractable time even on complex problems. It has been naturally used for feature selection (or in the case of microarray data, *gene selection*) and has shown promising performance [6,7,11,17–20,28,29] (for a survey of GA based methods on microarray analysis, the reader is referred to [15]). Unfortunately, due to the inherent nature of GA, it often takes a long time to locate the local optimum in a region of convergence and may sometimes not find the optimum with sufficient precision. One way to solve this problem is to hybridize GA with some memetic operations (also known as local search operations) [30–34] which are capable of fine-tuning and improving the solutions generated by the GA to make them more accurate and efficient. In diverse contexts, this form of evolutionary algorithms are referred to as Memetic algorithms (MAs), hybrid Evolutionary Algorithms (EAs), Baldwinian EAs, Lamarckian EAs, cultural algorithms or genetic local search. Recent studies on MAs have revealed their successes on a wide variety of real world problems. Particularly, they not only converge to high quality solutions, but also search more efficiently than their conventional counterparts [30–34].

In this study, we present an MA using possible synergy between filter and GA wrapper methods, particularly, a novel Markov blanket embedded GA (MBEGA) feature selection algorithm for cancer classification problem. MBEGA

uses Markov blanket to fine-tune the search by adding the relevant features or removing the redundant and/or irrelevant features in the GA solutions. This memetic algorithm takes advantage of both Markov blanket and GA wrapper feature selection with the aim to improve classification performance and accelerate the search to retain relevant and remove redundant features. We take representative methods from each of filter, wrapper, and standard GA and show that MBEGA's performance is comparable to or superior than the existing counterparts in terms of classification accuracy, number of selected features, computational cost, and/or robustness.

The rest of this paper is organized as follows. Section 2 describes our proposed algorithm MBEGA. Section 3 presents experimental results and discussions on four synthetic datasets and eleven microarray datasets. Finally, Section 4 concludes this study.

2 System and Methodology

In this section, we present an overview of the Markov Blanket, Approximate Markov Blanket and our proposed Markov Blanket-Embedded GA Feature Selection (MBEGA), which is a hybrid of filter and GA wrapper methods.

2.1 Markov Blanket

In 1996, Koller and Sahami [37] proposed a cross-entropy based technique, known as Markov Blanket for identifying redundant and irrelevant features. If F is the full set of features and C the class, the Markov blanket of a feature F_i is defined as follows:

Definition: (Markov Blanket) Let M be a subset of features which does not contain F_i, i.e., $M \subseteq F$ and $F_i \notin M$. M is a Markov blanket of F_i if F_i is conditionally independent of $(F \cup C) - M - \{F_i\}$ given M, i.e., $P(F - M - \{F_i\}, C|F_i, M) = P(F - M - \{F_i\}, C|M)$.

Two attributes A and B are conditionally independent given X, if $P(A|X, B) = P(A|X)$, that is B gives no information about A beyond what is already in X. If a feature F_i has a Markov blanket M within the currently selected feature subset, it suggests that F_i gives no more information beyond M about C and other selected features, therefore, F_i could be removed safely. However, since the computational complexity to determine the conditional independence of features is typically very high, Yu and Liu [3,27] considered using only one feature to approximate the Markov blanket of F_i:

Definition: (Approximate Markov Blanket) For two features F_i and $F_j (i \neq j)$, F_j is said to be an approximate Markov blanket of F_i if $SU_{j,C} \geq SU_{i,C}$ and $SU_{i,j} \geq SU_{i,C}$ where the symmetrical uncertainty SU [35] measures the correlation between features (including the class, C). It is defined as:

$$SU(F_i, F_j) = 2\left[\frac{IG(F_i|F_j)}{H(F_i) + H(F_j)}\right] \tag{1}$$

where $IG(F_i|F_j)$ is the information gain [36] between features F_i and F_j, $H(F_i)$ and $H(F_j)$ denote the entropies of F_i and F_j respectively. $SU_{i,C}$ denotes the correlation between feature F_i and the class C, and is named *C-correlation*. A feature is thus considered to be relevant if its *C-correlation* is higher than some given threshold γ which is user-specific, i.e., $S_{i,C} > \gamma$. A relevant feature without any approximate Markov blanket is called a predominant feature.

2.2 Markov Blanket-Embedded GA

In this section, we give details on the proposed memetic algorithm, particularly, the Markov Blanket-Embedded GA (MBEGA). The pseudo code of the MBEGA is outlined in Figure 2.

Markov Blanket Embedded Genetic Algorithm (MBEGA)
BEGIN
(1) **Initialize:** Randomly generate an initial population of feature subsets encoded with binary string.
(2) **While**(*not converged or computational budget is not exhausted*)
(3) Evaluate fitness of all feature subsets in the population based on $J(S_c)$.
(4) Select the elite chromosome c_b to undergo Markov blanket based memetic operation.
(5) Replace c_b with improved new chromosome c_b'' using Lamarckian learning.
(6) Perform evolutionary operators based on restrictive selection, crossover, and mutation.
(7) **End While**
END

Fig. 2. Markov blanket embedded genetic algorithm (MBEGA) for gene selection

At the start of the MBEGA search, an initial GA population is initialized randomly with each chromosome encoding a candidate feature subset. In the present work, each chromosome is composed of a bit string of length equal to the total number of features in the feature selection problem of interest.

Using binary encoding, a bit of '1' ('0') implies the corresponding feature is selected (excluded). The fitness of each chromosome is then obtained using an objective function based on the induction algorithm:

$$Fitness(c) = J(S_c) \tag{2}$$

where S_c denotes the selected feature subset encoded in a chromosome c, and the feature selection objective function $J(S_c)$ evaluates the significance for the given feature subset S_c. Here, $J(S_c)$ is the generalization error obtained for S_c which can be estimated using cross validation or bootstrapping techniques. Note that when two chromosomes are found having similar fitness (i.e., (i.e. for a misclassification error of less than one data instance, the difference between their fitness is less than a small value of $\varepsilon = 1/n$, where n is the number of instances), the one with a smaller number of selected features is given higher chance of surviving to the next generation.

In each GA generation, the elite chromosome, i.e., the one with the best fitness value then undergoes Markov blanket based memetic operators/local search in the spirit of Lamarckian learning [31,32]. The Lamarckian learning forces the genotype to reflect the result of improvement through placing the locally improved individual back into the population to compete for reproductive opportunities. Two memetic operators, namely an *Add* operator that inserts a feature into the elite chromosome, and a *Del* operator that removes existing features from the elite chromosome, are introduced in the MEBGA. The important question is which feature to add and which feature to delete. Ideally, each deleted feature should be covered by some other selected feature in the existing GA solution. This requirement is fast fulfilled by using Markov blanket concept.

We illustrate our method here. For a given selected subset encoded in the chromosome c, we define $\mathbf{X}$ and $\mathbf{Y}$ as the sets of selected and excluded features encoded in c, respectively. The purpose of the *Add* operator is to select a highly correlated feature Y_i using *C-correlation* measure from $\mathbf{Y}$ and moves it to $\mathbf{X}$. The *Del* operator on the other hand serves to select highly correlated features X_i from $\mathbf{X}$ and remove other features that are covered by X_i using approximate Markov blanket. If there is no feature in the approximate Markov blanket of X_i, the operator also tries to delete X_i itself. The details of these two memetic operators are outlined in Figure 3 and Figure 4.

It is worth noting that the *C-correlation* measure of each feature (i.e., action (1) in Figure 3 and Figure 4) only needs to be calculated once. This feature ranking information is then archived for use in *Add* and *Del* operations, in fine-tuning the GA solutions throughout the entire search. We further illustrate the processing of *Add* and *Del* operations using Figure 5. Here, $F5$ and $F4$ represent the highest and the lowest ranked features in $\mathbf{Y}$ while $F3$ and $F6$

***Add* Operator:**
BEGIN
(1) Rank the features in $\mathbf{Y}$ in a descending order based on *C-correlation* value.
(2) Select a feature Y_i in $\mathbf{Y}$ using linear ranking selection [38] so that the larger the *C-correlation* of a feature in $\mathbf{Y}$ the more likely it will be selected.
(3) Add Y_i to $\mathbf{X}$.
END

Fig. 3. *Add* operation

***Del* Operator:**
BEGIN
(1) Rank the features in $\mathbf{X}$ in a descending order based on *C-correlation* value.
(2) Select a feature X_i in $\mathbf{X}$ using linear ranking selection [38] so that the larger the *C-correlation* of a feature in $\mathbf{X}$ the more likely it will be selected.
(3) Eliminate all features in $\mathbf{X} - \{X_i\}$ which are in the approximate Markov blanket of X_i. If no feature is eliminated, remove X_i itself.
END

Fig. 4. *Del* operation

are the highest and the lowest ranked features in $\mathbf{X}$, moreover, $F3$ is the approximate Markov blanket of $F1$ and $F6$. In the *Add* operation, $F5$ is the most likely feature to be moved to $\mathbf{X}$. While in the *Del* operation, $F3$ is the most likely feature to be selected in $\mathbf{X}$. Hence, $F1$ and $F6$ will be deleted since they are covered by $F3$. The two most probable resultant chromosomes after the *Add* and *Del* operations are also depicted in Figure 5.

It is possible to quantify the computational complexity of the two memetic operators based on the search range L, which defines the maximum numbers of both *Add* and *Del*. Therefore, with L possible *Add* operations and L possible *Del* operations, there are a total of L^2 possible combinations of *Add* and *Del* operations applied on a chromosome. The L^2 combinations of *Add* and *Del* are applied to the candidate chromosome in a random order and the procedure stops once an improvement is obtained either in terms of fitness or reduction in the number of selected features without deterioration in the fitness value. The procedure of the Markov blanket based memetic operation applied on the elite chromosome of each GA search generation is outlined in Figure 6.

After applying the above Lamarckian learning process on the elite chromo-

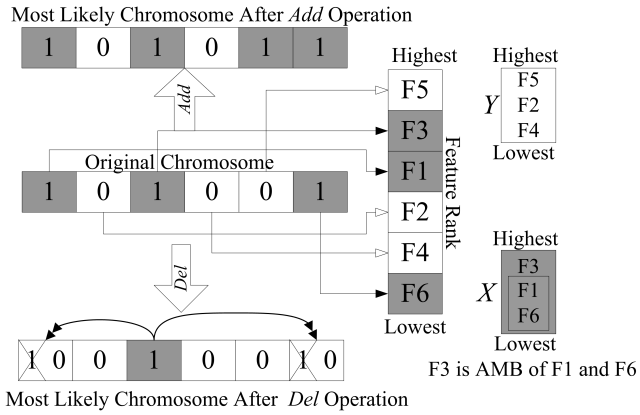

Fig. 5. Markov blanket based memetic operations (AMB denotes Approximate Markov Blanket)

Markov Blanket Based Memetic Operation
BEGIN

(1) Select the elite chromosome c_b to undergo memetic operations.
(2) **For** $l = 1$ **to** L^2
(3) Generate a unique random pair $\{a, d\}$ where $0 \leq a, d < L$.
(4) Apply a times Add on c_b to generate a new chromosome c'_b.
(5) Apply d times Del on c'_b to generate a new chromosome c''_b.
(6) Calculate fitness of modified chromosome c''_b based on $J(S_c)$.
(7) **If** c''_b is better than c_b either on fitness or number of features
(8) Replace the genotype c_b with c''_b and stop memetic operation.
(9) **End If**
(10) **End For**

END

Fig. 6. Markov blanket based memetic operators

some, the GA population then undergoes the usual evolutionary operations including linear ranking selection[38], uniform crossover, and mutation operators with elitism[26]. However, if prior knowledge on the optimum number of features is available, we permit the incorporation of such information in our proposed memetic algorithm by constraining the number of bits '1' in each chromosome to a maximum of m (m is lightly greater than the optimum number of features) in the evolutionary search process. To do so, we proposed using restrictive crossover and mutation [34] instead of the basic GA evolutionary operators, so that the number of bits '1' in each chromosome does not violate the constraint posed by the prior knowledge on m through the search.

3 Empirical Study

In this section, we study the performance of the proposed method against recent filter and wrapper feature selection algorithms using both high-dimensional synthetic problems and real world microarray data. In particular, we consider the FCBF [3,27], BIRS [39] and standard GA feature selection algorithms to compare the proposed MBEGA method. These algorithms have been successfully used for gene selection and demonstrated to attain promising performance [3,6,7,39]. Some brief overviews of these algorithms are provided here.

FCBF, proposed by Yu and Liu [3,27], is a fast correlation-based filter method. It begins by selecting a subset of relevant features whose *C-correlation* are larger than a given threshold γ, and then sorts the relevant features in descending order in terms of *C-correlation*. Using the sorted feature list, redundant features are eliminated one-by-one in a descending order. A feature is redundant only if it has an approximate Markov blanket. The remaining feature subset thus contains the predominant features with zero redundant features in terms of *C-correlation*.

BIRS (Best Incremental Ranked Subset) [39] uses a similar scheme as the FCBF but evaluates the goodness of features using a classifier. Like FCBF, BIRS begins by ranking the genes according to some measure of interest and then sequentially selects the ranked features one-by-one based on their incremental usefulness. Given the current selected feature subset and the learning algorithm, a feature is incrementally useful if its addition to the selected feature subset results in statistically significant improvements of classification accuracy. Hence, after the initial ranking, BIRS calls the classifier as many times as the number of features. Based on the ranking measure used in the first step, BIRS is categorized either as BIRS_F or BIRS_W, which rank features based on *C-correlation* (i.e. symmetrical uncertainty between feature F_i and the class C) or individual predictive power, respectively. In [39], BIRS_F has been shown to obtain competitive classification accuracy and selected similar number of features but in less time compared to BIRS_W. Hence, we choose BIRS_F as the representative method in the present study. Note that the use of BIRS_F also provides a fair comparison with FCBF and MBEGA, since all of these methods use the same feature ranking measure.

GA feature selection is similar to MBEGA except that it does not involve any memetic operators. Both GA and MBEGA use the same parameter setting of population size = 50, crossover probability = 0.6, and mutation rate = 0.5. MBEGA or GA search stops when convergence to the global optimal has occurred or the maximum computational budget allowable (i.e., 2000 fitness functional calls) is reached. It is worth noting that the fitness function calls made to $J(S_c)$ in the memetic operations are also included as part of the

total fitness function calls. The memetic operation range L in MBEGA is empirically set to 4. These configurations are kept consistent in our study on both the synthetic and microarray data.

FCBF, BIRS, and the classifiers (C4.5 and SVM) used in the following studies have been developed using Weka environment [40]. The parameters for FCBF, C4.5, and SVM were based on the defaults in Weka. For BIRS, the configurations reported in [39] are used.

3.1 Empirical Results on Synthetic Data

We first present the results of our study on four synthetic datasets. We take an often used dataset Corral [41] and extend it with additional redundant and irrelevant features. Our purpose is to evaluate the four algorithms on the various aspects of redundant and irrelevant features. The algorithms are expected to perform well on some aspects but not all aspects of redundant and irrelevant features.

The first dataset is the Corral data [41], which consist of 6 boolean features $(A0, A1, B0, B1, I, R75)$ and a boolean class C defined by $C = (A0 \wedge A1) \vee (B0 \wedge B1)$. The features $A0, A1, B0$ and $B1$ are independent to each other, feature I is uniformly random and irrelevant to class C. $R75$ matching 75% of C is redundant. The second data, Corral-46 generated in [27], is obtained by introducing more irrelevant and redundant features to the original Corral data. It includes the optimal feature subset $(A0, A1, B0, B1)$, 14 irrelevant features, and 28 additional redundant features. Among the 28 additional redundant features, for each $A0, A1, B0$ and $B1$, there are 7 features match with it at levels of $1, 15/16, 14/16, \ldots, 10/16$. The third data, Corral-50, is the same as Corral-46 except that it includes redundant features $R75, R80, R85$ and $R90$, which respectively match C at the level of $75\%, 80\%, 85\%$ and 90% (In the following text, we concisely denote the subset of $R75, R80, R85$, and $R90$ as $R+$). $R+$ are all highly correlated to C and could have larger *C-correlation* measure than those of features in optimal feature subset. To test the methods on problem of high-dimension and high-redundancy like the case on microarray data, we generate the last data sets Corral-10000, which are similar to Corral-50 but including much more (up to 9964) irrelevant features.

The four synthetic datasets considered here are also summarized in Table 1. In the present study, we consider C4.5 as the classifiers for feature subset evaluation since it provides the optimal subset of $(A0, A1, B0, B1)$ at a theoretical prediction accuracy of 100%.

The feature selection performances of FCBF, BIRS, GA and MBEGA on the synthetic datasets using ten 10-fold cross-validation with C4.5 classifier

Table 1
Summary of the four synthetic datasets

Dataset	Features
Corral	**OS:** $A0, A1, B0, B1$
	RF: $R75$
	IF: I
Corral-46	**OS:** $A0, A1, B0, B1$
	RF: $A0_{\{1,15/16,...,10/16\}}$, $A1_{\{1,15/16,...,10/16\}}$, $B0_{\{1,15/16,...,10/16\}}$, $B1_{\{1,15/16,...,10/16\}}$
	IF: $I0, I1, \ldots, I13$
Corral-50	**OS:** $A0, A1, B0, B1$
	RF: $A0_{\{1,15/16,...,10/16\}}$, $A1_{\{1,15/16,...,10/16\}}$, $B0_{\{1,15/16,...,10/16\}}$, $B1_{\{1,15/16,...,10/16\}}$ $R90, R85, R80, R75$
	IF: $I0, I1, \ldots, I13$
Corral-10000	**OS:** $A0, A1, B0, B1$
	RF: $A0_{\{1,15/16,...,10/16\}}$, $A1_{\{1,15/16,...,10/16\}}$, $B0_{\{1,15/16,...,10/16\}}$, $B1_{\{1,15/16,...,10/16\}}$ $R90, R85, R80, R75$
	IF: $I0, I1, \ldots, I9963$

OS: optimal subset; **RF:** redundant features; **IF:** irrelevant features.

$A0_{\{1,15/16,...,10/16\}}$ denotes the subset of 7 features matching $A0$ at levels of 1, 15/16, 14/16,. . . ,10/16. (Similar definition applied to $A1_{\{1,15/16,...,10/16\}}$, $B0_{\{1,15/16,...,10/16\}}$, $B1_{\{1,15/16,...,10/16\}}$)

Table 2
Feature selection by each algorithm on synthetic data

		FCBF	BIRS	MBEGA	GA
Corral	S_c	(R75,A0,A1,B0,B1)	(R75)	**(A0,A1,B0,B1)**	**(A0,A1,B0,B1)**
	$\overline{\lvert S_c \rvert}$	5	1	4 ± 0	4 ± 0
	acc	96.02	75.00	100.00 ± 0	100.00 ± 0
Corral-46	S_c	**(A0,A1,B0,B1)**	**(A0,A1,B0,B1)**	**(A0,A1,B0,B1)**	(. . .)
	$\overline{\lvert S_c \rvert}$	4	4	4 ± 0	12.4 ± 0.4
	acc	100.00	100.00	100.00 ± 0	100.00 ± 0
Corral-50	S_c	(R+,A0,A1,B0,B1)	(R90)	**(A0,A1,B0,B1)**	(. . .)
	$\overline{\lvert S_c \rvert}$	8	1	4 ± 0	16.3 ± 1.2
	acc	97.97	90.63	100.00 ± 0	100.00 ± 0
Corral-10000	S_c	(R+,A0,A1,B0,B1)	(R90)	**(A0,A1,B0,B1)**	(. . .)
	$\overline{\lvert S_c \rvert}$	8	1	4 ± 0	13.6 ± 2.5
	acc	97.97	90.63	100.00 ± 0	82.51 ± 4.72

S_c: selected feature subset; $\overline{|S_c|}$: average number of selected features; acc: classification accuracy.

are reported in Table 2. In particular, the selected feature subset, number of selected features, and average classification accuracy are tabulated. Due to the stochastic nature of MBEGA and GA, the average feature selection results of MBEGA and GA for ten independent runs are reported. The maximum number of selected features in each chromosome, m, is set to 50.

The results in Table 2 show that the first three methods select optimal feature subset and obtain a classification accuracy of 100% on Corral-46 dataset. On the other three datasets, only MBEGA successfully identify the optimal

feature subset and generating a perfect accuracy of 100%. FCBF also identifies $A0, A1, B0, B1$ but fails to eliminate features from $R+$. Because $R+$ have higher *C-correlation* than those of optimal feature subset, it is impossible for FCBF to find approximate Markov Blanket of them and hence unable to get rid of them. BIRS select only one feature with biggest *C-correlation*. For instance, on dataset Corral-50, BIRS selected only the top ranked feature, $R90$, because using this single feature a classification accuracy of 90.63% can be obtained and the addition of other features dose not make significant improvements. Since BIRS is a sequential forward search method that is incapable of considering interactions between features, it is more likely to get suboptimal results. GA select much more features than other methods. Without Markov blanket based memetic operators, GA individually is not efficient in reducing feature size.

3.2 Experimental Results on Microarray Data

In this section, we consider some real world microarray datasets having significantly large number of features (genes). In particular, 11 publicly available datasets in [10,42] are considered in the present study. A brief overview of these 11 datasets are summarized in Table 3.

Table 3
Description of 11 Microarray Datasets

Dataset	#Total Genes (T)	#Instances (n)	#Classes
Colon Tumor	2000	60	2
Central Nervous System	7129	60	2
ALL-AML	7129	72	2
Breast Cancer	24481	97	2
Lung Cancer	12533	181	2
Ovarian Cancer	15154	253	2
ALL-AML-3	7129	72	3
ALL-AML-4	7129	72	4
Lymphoma	4026	62	3
MLL	12582	72	3
SRBCT	2308	83	4

In many earlier works, researchers typically split the original dataset into two sets, a training set and a test set in a random fashion. Gene selection is then performed on the training set and the goodness of selected genes is assessed from the unseen test set. However, due to the small number of instances, such an approach is now recognized by the community as unreliable. Instead, Ambroise and McLachlan [8] suggested to split the data using external (10-fold) cross validation or .632+ bootstrap. Furthermore, a comparison of various error estimation methods on microarray classification [43] also suggests that

.632+ bootstrap may be more appropriate than other estimators including resubstitution estimator, k-fold cross-validation, and leave-one-out estimation. Taking this cue, we employed a balanced external .632+ bootstrap to evaluate the performances of the feature selection algorithms considered in this study. The .632+ bootstrap involves sampling a training set with replacement from the original dataset. The test set is formed by those samples omitted from the training set. The .632+ bootstrap is repeated K times and the final bootstrap error estimator $\epsilon_{b.632}$ is defined as:

$$\epsilon_{b.632} = \frac{1}{K}\sum_{i=1}^{K}(0.368\alpha_i + 0.632\beta_i) \tag{3}$$

where α_i and β_i are the training error and test error on the i^{th} resampling. Following the work in [8], the bootstrap samples are formed with $K = 30$ replicates. Each instance in the original dataset is made to appear exactly 30 times in the balanced bootstrap training samples. Feature selection is then performed using only the training samples. In BIRS, MBEGA and GA, a cross validation scheme is applied on the training samples for feature subset evaluation. Finally, the test error is estimated on the unseen test samples. The classification accuracy is then estimated using Equation (3).

In the present study, the support vector machine (SVM) is chosen as the classifier since it has been demonstrated in the literature to outperform many existing machine learning methods (e.g., KNN, C4.5, Naive Bayes, etc) on two-class [9] or multi-class [10,21] microarray classification problems. In MBEGA and GA, the SVM radius margin bound [44] is used to estimate the generalization error $J(S_c)$ with only a single run of the training dataset. For BIRS, SVM with 5-fold cross validation is used for estimating the classification accuracy, i.e., the evaluation procedure follows that described in [39] to avoid changes to the original algorithm proposed. As previous studies have considered no more than 50 and 150 genes as optimal feature subset on two-class datasets [2–5,8,9,14,16,19,20] and multi-class datasets [10,17,18], respectively. We constrain the maximum number of selected features m in each chromosome to these bounds in both MBEGA and GA. Because the *C-correlation* measure is not designed for continuous data, we use Fayyad and Irani's Minimum Description Length (MDL) method [45] to discretize the microarray data as a pre-processing step.

Table 4 presents the average classification accuracy, average number of selected genes, and average running time for each feature selection algorithm on the eleven datasets over 30 runs of .632+ bootstraps.

Table 4
Performance of feature selection algorithms on microarray datasets

		FCBF	BIRS	MBEGA	GA
Colon Tumor	*acc*	84.54 ± 5.63	76.49 ± 7.50	$\mathbf{85.66 \pm 5.46}$	81.01 ± 7.58
(2000×60)	$\overline{\lvert S_c \rvert}$	21.0 ± 2.6	1.8 ± 0.9	24.5 ± 7.0	23.3 ± 3.6
	t(s)	1.1	56.0	70.6	66.8
Central Nervous System	*acc*	$\mathbf{73.29 \pm 6.30}$	64.14 ± 5.07	72.21 ± 5.91	68.30 ± 5.85
(7129×60)	$\overline{\lvert S_c \rvert}$	52.8 ± 7.9	1.7 ± 1.0	20.5 ± 6.9	24.1 ± 3.2
	t(s)	4.2	36.6	81.1	78.5
ALL-AML	*acc*	93.82 ± 8.37	89.34 ± 6.15	$\mathbf{95.89 \pm 2.46}$	92.31 ± 3.37
(7129×72)	$\overline{\lvert S_c \rvert}$	32.8 ± 18.1	2 ± 0.8	12.8 ± 4.9	25.2 ± 3.3
	t(s)	5.6	378.2	112.3	118.0
Breast Cancer	*acc*	75.89 ± 4.80	67.98 ± 6.55	$\mathbf{80.74 \pm 3.45}$	68.81 ± 5.80
(24481×97)	$\overline{\lvert S_c \rvert}$	127.9 ± 10.4	2.3 ± 1.0	14.5 ± 4.2	22.1 ± 4.3
	t(s)	28.3	3766.4	266.9	243.8
Lung Cancer	*acc*	95.70 ± 6.69	97.31 ± 1.86	$\mathbf{98.96 \pm 0.88}$	98.06 ± 1.12
(12533×181)	$\overline{\lvert S_c \rvert}$	66.3 ± 43.0	2.5 ± 0.7	14.1 ± 7.0	24.4 ± 3.2
	t(s)	30.8	1273.1	1041.7	1025.5
Ovarian Cancer	*acc*	$\mathbf{99.91 \pm 0.30}$	98.50 ± 1.67	99.71 ± 0.53	99.43 ± 0.40
(15154×253)	$\overline{\lvert S_c \rvert}$	31.3 ± 4.0	2.0 ± 0.7	9.0 ± 2.6	23.3 ± 2.8
	t(s)	26.7	2098.9	2689.5	2588.5
ALL-AML-3	*acc*	95.76 ± 2.94	84.49 ± 8.72	$\mathbf{96.64 \pm 2.71}$	90.96 ± 4.86
(7129×72)	$\overline{\lvert S_c \rvert}$	77.7 ± 9.9	2.4 ± 1.0	18.1 ± 5.7	75.1 ± 6.4
	t(s)	3.7	1175.4	176.6	174.2
ALL-AML-4	*acc*	$\mathbf{92.38 \pm 5.38}$	77.60 ± 9.16	91.93 ± 4.32	88.06 ± 5.82
(7129×72)	$\overline{\lvert S_c \rvert}$	99.3 ± 14.8	2.2 ± 1.3	26.2 ± 8.7	74.4 ± 5.9
	t(s)	4.4	1329.8	234.3	218.4
Lymphoma	*acc*	94.85 ± 9.33	86.11 ± 8.64	97.68 ± 2.81	$\mathbf{98.99 \pm 2.10}$
(4026×62)	$\overline{\lvert S_c \rvert}$	48.5 ± 26.6	2.0 ± 0.9	34.3 ± 8.0	74.9 ± 5.7
	t(s)	2.1	287.8	142.6	139.1
MLL	*acc*	$\mathbf{95.64 \pm 8.62}$	84.00 ± 8.67	94.33 ± 3.31	92.22 ± 4.47
(12582×72)	$\overline{\lvert S_c \rvert}$	101.6 ± 22.7	3.7 ± 1.2	32.1 ± 10.6	75.4 ± 5.8
	t(s)	8.4	2593.8	182.1	165.0
SRBCT	*acc*	98.94 ± 1.44	86.66 ± 7.59	$\mathbf{99.23 \pm 1.15}$	95.77 ± 3.22
(2308×83)	$\overline{\lvert S_c \rvert}$	98.6 ± 9.8	4.1 ± 1.2	60.7 ± 11.7	78.4 ± 5.6
	t(s)	2.2	386.8	246.2	232.6

acc: classification accuracy; $\overline{\lvert S_c \rvert}$: average number of selected genes; t(s): running time (second).

3.2.1 Classification Accuracy

In table 4, the best average classification accuracy among the four algorithms on each dataset is highlighted in bold typeface. Overall, MBEGA, FCBF and GA are observed to produce the best accuracy on 6/11, 4/11 and 1/11 datasets, respectively. Statistical test using random permutation test [2] [21,46] at signif-

[2] Random permutation test does not rely on independence assumptions.

icance level of 0.05 shows that MBEGA outperforms GA in terms of accuracy on all datasets except the Lymphoma dataset. The accuracies of FCBF and MBEGA are not significantly different from each other on 8/11 datasets. Out of the remaining 3 datasets (Breast Cancer, Lung Cancer, and Ovarian Cancer), MBEGA performs better than FCBF on the Breast Cancer and Lung Cancer datasets, while poorer on the Ovarian Cancer dataset. BIRS is observed to have the lowest classification accuracy among the four methods on all 11 datasets.

3.2.2 Number of Selected Genes

With respect to the number of selected genes, BIRS selects the smallest feature subset with no more than 4 genes on all datasets but it does so at the cost of low classification accuracy. On the other hand, MBEGA also selects a smaller number of gene features but with a classification accuracy that is competitive or superior to both FCBF and GA. Since FCBF selects predominant genes based on correlation measure and ignores the learning model in the inductive algorithm, some of the selected genes are actually redundant as they do not contribute to any improvement in the classification accuracy. In contrast, GA identifies suitable gene subsets solely based on the predictive ability of inductive algorithm. MBEGA on the other hand takes account of both correlation redundancy among genes and their cooperative classification ability via a wrapper scheme. Consequently it is able to eliminate more redundant features without deteriorating the classification accuracy and select genes more suitable for the inductive algorithm. The results in Table 4 suggest that on most of the datasets MBEGA obtains competitive or superior classification accuracy to FCBF using only one-eighth to one-half number of genes selected by FCBF. MBEGA also selects much fewer genes than GA but with significantly better accuracy. Especially on the multi-class datasets, MBEGA selects only about one-third the number of genes selected by GA. We also note that FCBF has a larger standard deviation in the number of selected features than MBEGA, particularly for ALL-AML, Breast Cancer, Lung Cancer, ALL-AML-4, Lymphoma, and MLL datasets, which suggests that MBEGA is more robust on different bootstrap resampling datasets than FCBF. The robustness of the various methods are further investigated later in Section 3.2.4.

3.2.3 Average Running Time

The average running time of the algorithms are reported in Table 4. The Pearson's correlation coefficient between the running time and other external factors (i.e., number of instances and number of classes, and total number of genes) are summarized in Table 5.

Table 5
Correlation coefficient between running time and other external factors

Correlation coefficient	FCBF	BIRS	MBEGA	GA
t(s) vs. #Instances	0.792	0.351	**0.969**	**0.970**
t(s) vs. #Classes	-0.500	-0.106	-0.296	-0.300
t(s) vs. Total # Genes	0.839	**0.916**	0.384	0.380

t(s): running time (second).

From table 4, we can see that the running time of the filter method FCBF are lesser than 31 seconds on all the datasets, which are slightly correlated with the number of instances and the total number of genes as shown in Table 5. FCBF is much faster than all other algorithms as it evaluates features without involving the time consuming classification. For the three wrapper methods, most of the time is spent on classification (i.e., fitness function call), hence other overhead (e.g., the ranking of *C-correlation* in BIRS and MBEGA, the evolutionary operations in MBEGA and GA) are considered negligible. The overall running time of BIRS , MBEGA and GA is correlated to the total number of classifications and the running time of a single classification, which are depended on some of the external factors (i.e., the number of instances, the number of classes, and the total number of genes). The running times of MBEGA and GA are highly correlated to the number of instances but uncorrelated to the total number of genes. This makes MBEGA more suitable for microarray data which is characterized with thousands of genes but with tens or hundreds of instances. MBEGA and GA run with identical classifier and total number of classifications, they consume similar running time in all the datasets. BIRS requires one classification for each gene, therefore its running time is mainly determined by the total number of genes, which is demonstrated in Table 5 with a high correlation coefficient of 0.916. The results in Table 4 show that BIRS takes around the same amount of running time as MBEGA and GA on 5/6 two-class datasets but when the number of genes increases tremendously, e.g., the Breast Cancer dataset has 24481 genes, the running time for BIRS increases to 3766.4 seconds which is 14 times more than that of MBEGA (266.9 seconds). BIRS also takes more time than MBEGA and GA on all the multi-class datasets.

3.2.4 Robustness

In the microarray datasets, the number of instances available for learning is typically small. As such the gene selection results obtained in the bootstrap re-sampling process are likely to have large variance. This is generally unacceptable in the point view of biologists, who believe that a good algorithm should be one that identifies the crucial genes for the purpose of diagnosis, therapeutics, or prognosis of cancer consistently and not by chance. For this reason, here we also investigate the robustness of different gene selection algorithms.

In this study, we determine the robustness using Z-score analysis [7,11]. Z-score indicates the significance of the frequency a gene getting selected. A gene with a high Z-score suggests that it is not selected by chance and an algorithm that identifies a gene subset containing genes of higher Z-score is deemed to be more robust. The Z-score of gene i is defined as:

$$Z = \frac{S_i - E(S_i)}{\sigma} \tag{4}$$

where S_i is the frequency that gene i is selected, $E(S_i)$ is the expected number of times gene i is selected, while σ is the standard deviation for S_i. Let $\overline{|S_c|}$ be the average number of selected genes, T be the total number of genes, then the probability of gene i being selected is denoted as, $P(S_i) = \overline{|S_c|}/T$. Whereupon, $E(S_i)$ and σ are calculated using $E(S_i) = P(S_i) \cdot K$ and $\sigma = \sqrt{P(S_i) \cdot (1 - P(S_i)) \cdot K}$ where K is the number of bootstrap replicates.

We sort the genes in a descending order in terms of their frequency of selection. The Z-score of the top 50 selected genes for all the four algorithms on the 11 microarray datasets are plotted in Figure 7 and Figure 8. The x-axis denotes the sorted top 50 most selected genes and the y-axis denotes the Z-score. The results indicate that the top 50 genes selected by MBEGA have larger Z-scores than that of the other counterpart algorithms, which suggests MBEGA is more robust than the other algorithms.

3.2.5 Overall Performance

From the empirical results obtained so far, it is worth noting that each method has its strengths and limitations. In particular, FCBF obtains good classification accuracy in the least amount of running time but at the expense of selecting many more genes; BIRS selects the least number of genes but suffers in terms of classification accuracy and also requires more running time than others; MBEGA attains best accuracy and robustness in a reasonable time. For different purposes, different methods could be selected. However in practice, a method with better overall performance on all the evaluation criteria is favorable. To quantitatively investigate the overall performance, a rank (i.e, 1,2,3,4) of evaluation criterion is assigned to each algorithm on a given dataset. If there is no significant difference between multiple algorithms on a given criterion, a average rank is distributed among them (e.g., on Colon Dataset, the classification accuracy is not significantly different for FCBF and MBEGA, so both of them them are assigned a rank of 1.5). The summations of ranks on all the 11 datasets are tabulated in Table 6. The smaller value indicates better performance of the corresponding algorithm. The best performance of each measure is highlighted with boldface. The summations of ranks for all the four evaluation criteria (shown in the last row of Table 6) suggest that

MBEGA has better overall performance than other methods.

Table 6
Rank summation of the four evaluation criteria

	FCBF	BIRS	MBEGA	GA
acc	19.5	41	**18.5**	31
$\overline{\lvert S_c \rvert}$	41.5	**11**	23.5	34
t(s)	**11**	38	30.5	30.5
Robustness	23	32	**11**	44
Sum	95	122	**83.5**	139.5

acc: classification accuracy; $\overline{\lvert S_c \rvert}$: average number of selected genes; t(s): running time (second).

3.2.6 Comparison with Other Methods in the Literature

Since MBEGA uses GA and SVM, we also compared it with other GA or SVM based gene selection methods. We compared the most related work in the literature using the same datasets.

Firstly the GA based methods are considered. In [6,7], Li and his colleagues proposed a GA/KNN method. The GA/KNN gene selection is repeatedly applied on a training data for 20,000 to 40,000 times and different gene subsets are selected. The most frequently selected genes are used for classification on an hold-out test data. In [6], GA/KNN is reported to obtained a classification accuracy of 97.06% using the top 50 genes on the ALL-AML dataset. On the same dataset, MBEGA attains a competitive accuracy of 95.89% with only 12.8 genes. Nevertheless, the .632+ bootstrap validation scheme applied in this study is more reliable than the hold-out scheme.

Two other GA based methods, GA/MLHD and GA/SVM are proposed in [17] and [18], respectively. Both of them are applied to the same multi-class NCI60 dataset with 1000 preselected genes and the results are compared in [18]. GA/MLHD is observed to attain a leave-one-out cross validation (LOOCV) accuracy of 70.73 using 12 selected genes and GA/SVM outperforms GA/MLHD with a LOOCV accuracy of 88.52% using 40 genes. In this study, we compared MBEGA to the superior method GA/SVM. It is worth noting that the accuracy reported in [18] are actually not evaluated on a test data which is independent of the gene selection procedure. In particular, the gene selection is conducted using the whole NCI60 dataset as a training data and LOOCV accuracy on the same data is reported as the final results. This so called internal cross validation [8,47] has been demonstrated to suffer from selection bias [8] or overfitting [47] problem, especially on datasets of small sample size. To alleviate this problem, an external bootstrap method as described earlier has been suggested for gene selection in [8,43]. Hence, to make a fair comparison, 30 runs of external .632+ bootstrap are applied for both MBEGA and GA/SVM using the NCI60 dataset. The results show that MBEGA attains

better accuracy of 70.38% using 47.6 selected genes, while GA/SVM obtains an accuracy of only 65.79% using 40 selected genes.

We further compared MBEGA with SVM based methods. In [4], Guyon and his colleagues proposed a gene selection method utilizing SVM based on recursive feature elimination (RFE). The SVM-RFE method attains 100% accuracy with 8 genes on ALL-AML data and 100% accuracy with 16 genes on Colon data under an internal cross validation, which has been demonstrated to suffer from selection bias [8]. The bias corrected results using external .632+ bootstrap in [8] show that the best accuracy of SVM-RFE on ALL-AML and Colon data are lower than 95% and 85% respectively, although it uses more than 64 genes. Zhou and Mao [16] proposed the other SVM based gene selection method using LS bound measure and sequential forward selection (SFS). The LS bound-SFS method using .632+ bootstrap is reported to obtain a best accuracy of 97.68% using 50 genes on ALL-AML data and a best accuracy of 84.95% using 46 genes on Colon data. For comparison, MBEGA reaches an accuracy of 95.89% on ALL-AML data and 85.66% on Colon data, which are better than that of SVM-RFE and competitive with that of LS bound-SFS. Nevertheless, MBEGA selects much less genes of 12.8 for ALL-AML and 24.5 for Colon.

4 Conclusions

In this paper, we have proposed a novel Markov blanket embedded Genetic Algorithm (MBEGA) for gene selection. We also take representative methods from each of filter (FCBF), wrapper (BIRS), and standard GA to study their strengths and weaknesses in the present work. Empirical study on 4 synthetic datasets and 11 microarray datasets suggest that MBEGA gives better overall performance than the existing counterparts in terms of all four evaluation criteria, i.e., classification accuracy, number of selected genes, computational cost, and robustness. The comparison to other methods in the literature also suggest MBEGA has competitive or better performance. MBEGA is capable of eliminating irrelevant and redundant features based on both Markov blanket and predictive power of wrapper model effectively, thus providing a small set of reliable genes for the biologists to conduct further study. Furthermore, by performing text mining on biological literature [18] and using prior biological knowledge from tools such as RSVP [48], MBEGA can help enhance the process of candidate biomarkers discovery and improve researchers' ability of leveraging the increasing amounts of publicly available research data. Hence, we expect MBEGA to serve as an excellent alternative for microarray analysis.

References

[1] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, and A. Levine, Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor an Normal Colon Tissues Probed by Oligonucleotide Arrays, Proc. Natl. Sci. USA, 96(12)(1999)6745-6750.

[2] T. R. Golub, D. K. Slonim, P. Tamayo, et al, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science, 286(5439)(1999)531-537.

[3] L. Yu and H. Liu, Redundancy based feature selection for miroarray data, In: Tech ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, USA, 22-25 Aug, 2004.

[4] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, Gene Selection for Cancer Classification Using Support Vector Machines, Machine Learning, 46(1-3)(2002)389-422.

[5] S. Dudoit, J. Fridlyand and T. P. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, Journal of the American Statistical Association 97 (2002)77-87.

[6] L. Li, L. G. Pedersen, T. A. Darden and C. R. Weinberg, Computational analysis of leukemia microarray expression data using GA/KNN method. Proceeding of the 1st Conference on Critical Assessment of Microarray Data Analysis, CAMDA2000,(2000).

[7] L. Li, C. R. Weinberg, T. A. Darden, and L. G. Pedersen, Gene Selection for Sample Classification Based on Gene Expression Data: Study of Sensitivity to Choice of Parameters of the GA/KNN Method, Bioinformatics, 17(12)(2001)1131-1142.

[8] C. Ambroise and G. McLachlan, Selection bias in gene extraction on the basis of microarray gene-expression data, Proc. Natl. Sci. USA, 99(2002)6562-6566.

[9] H. Liu, J. Li and L. Wong, A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. Genome Information,13(2002)51-60.

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

[11] T. Jirapech-Umpai and S. Aitken, Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. BMC Bioinformatics,2005,6:148.

[12] A. W. Liew, H. Yan and M. Yang, Pattern recognition techniques for the emerging field of bioinformatics: A review. Pattern Recognition, 38(11)(2005)2055-2073.

[13] T. Hsing, L. Liu, M. Brun and E. R. Dougherty, The coefficient of intrinsic dependence (feature selection using el CID). Pattern Recognition, 38(5)(2005)623-636.

[14] E. K. Tang, P. N. Suganthan and X. Yao, Gene selection algorithms for microarray data based on least squares support vector machine. BMC Bioinformatics,2006,7:95.

[15] M. Wahde and Z. Szallasi, A survey of methods for classification of gene expression data using evolutionary algorithms. Expert Review of Molecular Diagnostic,6(1)(2006)101-110.

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

[17] C. H. Ooi and P. Tan, Genetic algorithm applied to multi-class prediction for the analysis of gene expression data. Bioinformatics, 19(1)(2003)37-44.

[18] J. J. Liu, G. Cutler, W. Li, Z. Pan, S. Peng, T. Hoey, L. Chen and X. B. Ling, Multiclass cancer classification and biomarker discovery using GA-based algorithms. Bioinformatics, 21(11)(2005)2691-2697.

[19] L. Li, W. Jiang, X. Li, K. L. Moser, Z. Guo, L. Du, Q. Wang, E. J. Topol, Q. Wang and S. Rao, A robust hybrid between genetic algorithm and support vector machine for extracting an optimal featuer gene subset. Genomics, 85(2005)16-23.

[20] J. M. Deutsch, Evolutionary algorithms for finding optimal gene sets in microarray prediction. Bioinformatics 19(1)(2003)45-52.

[21] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin and S. Levy, A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics, 21(5)(2005)631-643.

[22] M. Dash and H. Liu, Feature selection for Classification, Intelligent Data Analysis,1(3)(1997)131-156.

[23] M. Dash and H. Liu, Consistency-based search in feature selection, Artificial Intelligence, 151(1-2)(2003)155-176.

[24] H. Liu and L. Yu, Toward Integrating feature selection algorithms for classification and clusterfing, IEEE Trans. Knowledge and Data Engineering, 17(4)(2005)491-501.

[25] R. Kohavi and G. H. John, Wrapper for Feature Subset Selection, Artificial Intelligence, 97(1-2)(1997)273-324.

[26] J. H. Holland, Adaptation in natural artificial systems. 2nd edition, MIT Press (1992)

[27] L. Yu and H. Liu, Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research, 5(2004)1205-1224.

[28] J. H. Yang and V. Honavar, Feature Selection Using a Genetic Algorithm, IEEE Intelligent Systems, 13(2)(1998)44-49.

[29] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, Dimensionality Reduction Using Genetic Algorithms, IEEE Trans. Evolutionary Computation, 4(2)(2000)164-171.

[30] P. Moscato, On evolution, search, optimization, genetic algorithms and martial arts: toward memetic algorithms, Tech. Rep. Caltech Concurrent Computation Program, Rep. 826, California Inst. Technol., Pasadena, CA, 1989.

[31] N. Krasnogor, Studies on the Theory and Design Space of Memetic Algorithms, Ph.D. Thesis, Faculty of Computing, Mathematics and Engineering, University of the West of England, Bristol, U.K, 2002.

[32] Y. S. Ong and A. J. Keane, Meta-Lamarckian in Memetic Algorithm, IEEE Trans. Evolutionary Computation, 8(2)(2004)99-110.

[33] Y. S. Ong, M. H. Lim, N. Zhu and K. W. Wong, Classification of Adaptive Memetic Algorithms: A Comparative Study, IEEE Transactions On Systems, Man and Cybernetics - Part B, 36(1)(2006)141-52.

[34] Z. Zhu, Y. S. Ong and M. Dash, Wrapper-Filter Feature Selection Algorithm Using A Memetic Framework, IEEE Transactions On Systems, Man and Cybernetics - Part B, accepted 2006.

[35] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, Numerical Recipes in C, Cambridge University Press, Cambridge, 1998.

[36] J. R. QuinLan, C4.5: Programs for Machine Learning, San Mateo, Morgan Kaufman, 1993.

[37] D. Koller and M. Sahami, Toward optimal feature selection, In 13th International Conference on Machine Learning, Morgan Kaufmann, Bari, Italy, 1996.

[38] J. E. Baker, Adaptive Selection Methods for Genetic Algorithms, In Proc. Int'l Conf. Genetic Algorithm and Their Applications, (1985)101-111.

[39] R. Ruiz, J. C. Riquelme and J. S. Aguilar-Ruiz, Incremental wrapper-based gene selection from microarray data for cancer classification. Pattern Recognition, 39(12)(2006)2383-2392.

[40] I. H. Witten and E. Frank, Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[41] G. H. John, R. Kohavi and K. Pfleger, Irrelevant feature and subset selection problem. In proceeding of the Eleventh International Conference on Machine Learning, 121-129,1994.

[42] J. Li and H. Liu, Kent Ridge Biomedical Data Set Repository, Http://sdmc-lit.org.sg/GEDatasets, 2002.

[43] U. M. Braga-Neto and E. R. Dougherty, Is cross-validation valid for small-sample microarray classification?, Bioinformatics,20(3)(2004)374-380.

[44] O. Chapelle, V. Vapnik, O. Bousqet and S. Mukherjee, Choosing Multiple Parameters for Support Vector Machines, Machine Learning, 46(1)(2002)131-159.

[45] U. Fayyad, K. Irani, Multi-interval discretization of continuous-valued attributes for classification learning. The 13th International Joint Conference on Atrificial Intelligence, Chambery, France, 1993.

[46] P. I. Good, Permutation Tests: A Pratical Guide to Resampling Methods for Testing Hypothese, 2nd ed. New York, Springer-Verlag, 2000

[47] J. Reunanen, Overfitting in making comparisons between variable selection methods, Journal of Machine Learning Research, 3(2003)1371-1382.

[48] M. Berens, H. Liu, L Yu, Fostering biological relevance in feature selection for microarray data, IEEE Intelligent Systems,20(6)(2005)71-73.

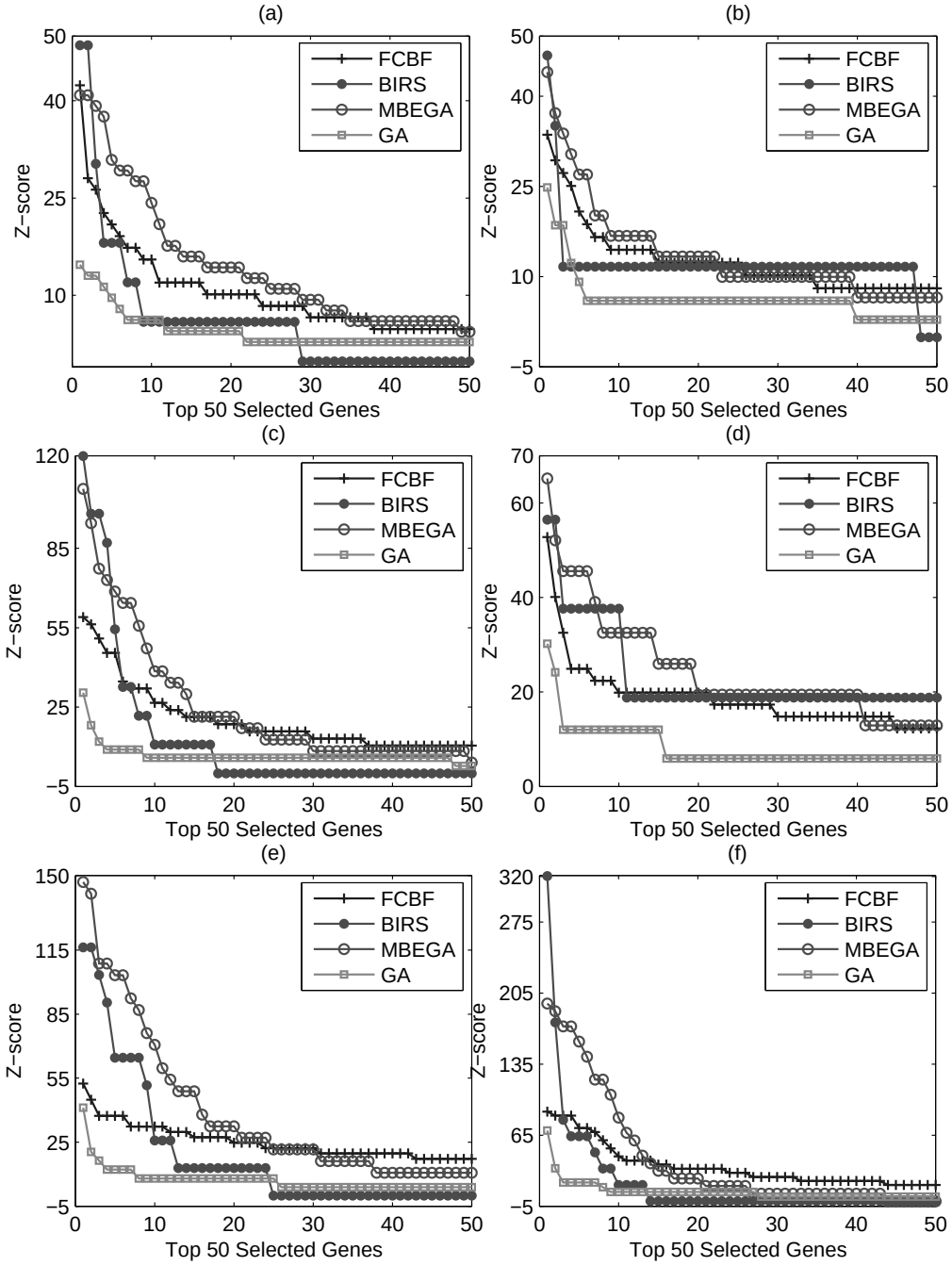

Fig. 7. The Z-score of the top 50 most frequently selected genes (a)Colon Cancer, (b)Central Nervous System, (c)ALL-AML, (d)Breast Cancer, (e)Lung Cancer, (f)Ovarian Cancer.

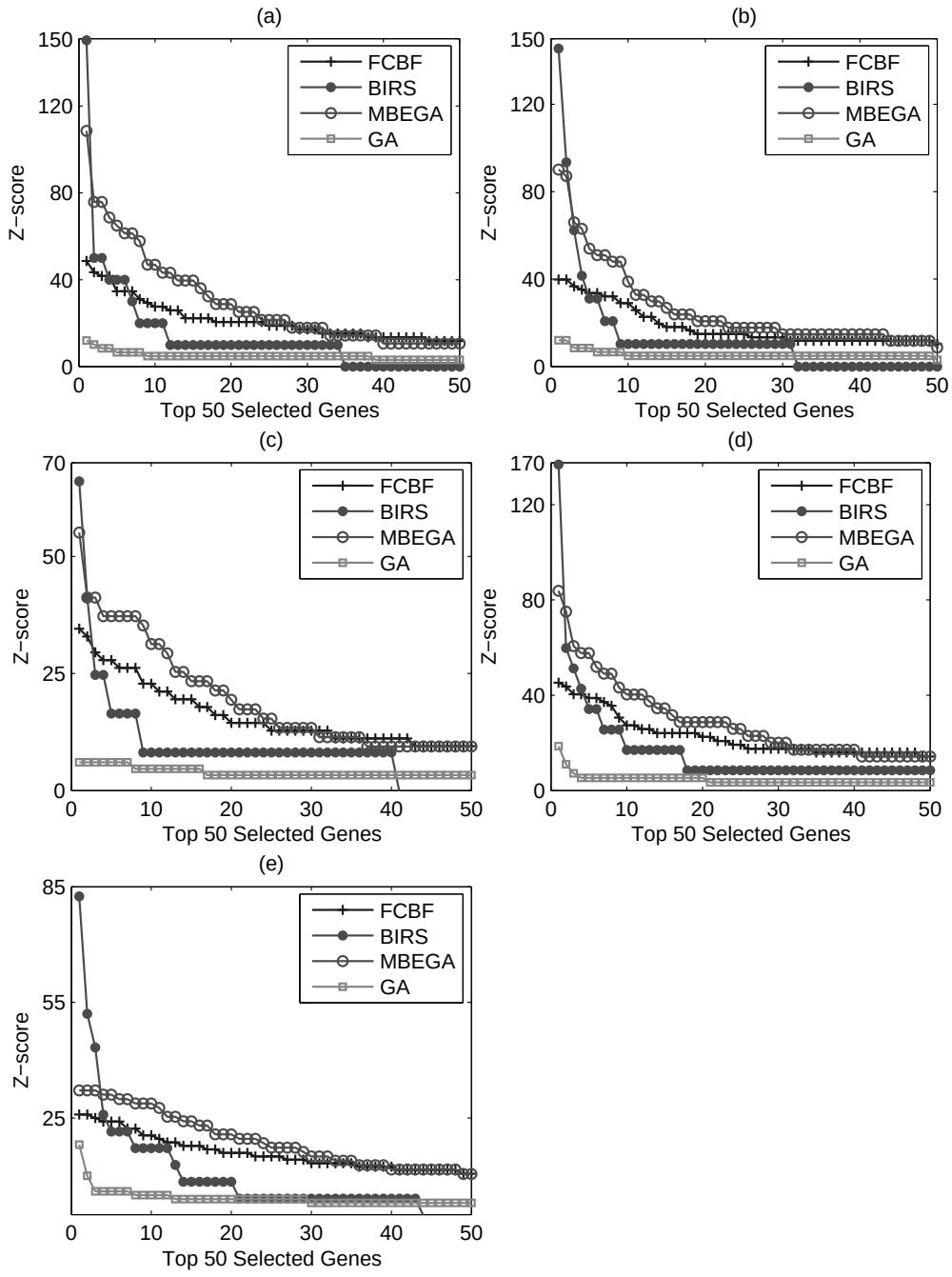

Fig. 8. The Z-score of the top 50 most frequently selected genes (a)ALL-AML-3, (b)ALL-AML-4, (c)Lymphoma, (d)MLL, (e)SRBCT.