Computers in Biology and Medicine 81 (2017) 11–23

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Wrapper-based gene selection with Markov blanket

Aiguo Wang[a], Ning An[a,*], Jing Yang[a], Guilin Chen[b], Lian Li[a], Gil Alterovitz[c,d,e]

[a] School of Computer and Information, Hefei University of Technology, Hefei, China
[b] School of Computer and Information Engineering, Chuzhou University, Chuzhou, China
[c] Center for Biomedical Informatics, Harvard Medical School, Boston, USA
[d] Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, USA
[e] Children's Hospital Informatics Program at the Harvard/MIT Division of Health Sciences and Technology, Boston, USA

ARTICLE INFO

Keywords:
Gene selection
Microarray data
Symmetric uncertainty
Markov blanket
Wrapper methods

ABSTRACT

Gene selection seeks to find a small subset of discriminant genes from the gene expression profiles. Current gene selection methods such as wrapper-based models mainly address the issue of obtaining high-quality gene subsets. However, they are considerably time consuming, due to the existence of irrelevant and redundant genes. In this study, we present an improved wrapper-based gene selection method by introducing the Markov blanket technique to reduce the required wrapper evaluation time. In addition, our method can identify targeting genes while eliminating redundant ones in an efficient way. We use ten publicly available microarray datasets to evaluate the proposed method. The results show that our method can handle gene selection effectively. Our experimental results also show that wrapper-based method combined with the Markov blanket outperforms other competing methods in terms of classification accuracy and time/space complexity.

1. Introduction

The rapid development and maturing of microarray technology enables researchers to measure the expression profiles of thousands of genes in a single experiment simultaneously [1], and the analysis of microarray data is a good alternative to the diagnosis of cancers and the discovery of disease biomarkers at the molecular level [2,3]. Accordingly, various statistical analysis methods and machine learning models have been utilized to analyze gene expression profiles, whereas the intrinsic nature of microarray data that are characterized by small sample sizes and high dimensionality largely hinders their meaningful applications in practice [4,5]. For example, in the diagnosis of cancer with microarray data, since the number of genes typically exceeds the number of available samples, classifiers that are directly constructed on such data may suffer from poor generalization capacity and weak robustness [6]. In addition, there are relevant studies suggesting that only a few discriminant genes are associated with a certain cancer but predictive for cancer diagnosis [7], and that the original gene space consists of a wealth of noisy and redundant genes, which deteriorates the performance of a classification model. Naïve Bayes, for example, is sensitive to redundant features, and nearest neighbor-based learners are susceptible to irrelevant features in handling classification problems [8]. One feasible way to mitigate this problem is to select a small subset of discriminant genes from original gene space using an effective gene selection method [9,10].

Feature selection, also known as gene selection in the context of microarray data, plays an important role in the analysis of gene expression profiles, ranging from cancer diagnosis and gene clustering to tumor subtype classification and disease gene discovery [5]. Feature selection is a process of finding a small subset of informative features that are relevant to a specific task by discarding irrelevant and redundant features [11]. Besides reducing the high dimensionality, feature selection offers a multitude of benefits, including reducing time costs in classifier training, enhancing the generalization capacity of the constructed classifier, and helping biologists understand the underlying biological mechanisms and biologically validate the drug targets efficiently [12,13]. According to the framework proposed by Dash and Liu [14], feature selection methods typically consists of two components: a feature subset generator module and an evaluator module. The former exploits a given search strategy to generate candidate feature subsets, while the latter evaluates the quality of a feature or a subset of features and feeds the evaluation information to the feature subset generator to guide the next-round search of candidate feature subset. In feature selection, establishing powerful evaluation criteria for measuring the goodness of a feature subset largely determines the quality of finally selected features. Depending on

* Correspondence to: Hefei University of Technology, Hefei Tunxi Road 193, Hefei 230009, China.
E-mail addresses: wangaiguo2546@163.com (A. Wang), ning.g.an@acm.org (N. An), jsjyj0801@163.com (J. Yang), glchen@chzu.edu.cn (G. Chen), llian@hfut.edu.cn (L. Li), gil_alterovitz@hms.harvard.edu (G. Alterovitz).

http://dx.doi.org/10.1016/j.compbiomed.2016.12.002
Received 14 September 2016; Received in revised form 17 November 2016; Accepted 2 December 2016

whether a classifier is used as the evaluation function, we can group existing feature selection methods into three categories: filter methods, wrapper methods, and embedded methods [15]. Filter methods are independent of a classification model and measure the quality of a feature subset using only the intrinsic properties of training samples, so they are flexible in combination with various classifiers and have lower computational complexity. Further, commonly used filter metrics include distance-, dependency-, consistency-, and information theory-based metrics [14,16,17]. Compared with other three metrics, feature selection methods with information theory have drawn much more attention because of their effectiveness and efficiency, and the capacity in reflecting the non-linear relationships among variables and capturing high order statistics of data. Correspondingly, researchers have proposed and developed a number of feature selectors on the basis of mutual information, such as symmetric uncertainty (SU), fast correlation based filter (FCBF), mutual information feature selection (MIFS), conditional mutual information maximum (CMIM), minimum redundancy maximum relevance (mRMR), and joint mutual information (JMI) [17]. In contrast to filter methods, wrapper methods use a specific classifier to evaluate the quality of a feature, and often use the classification accuracy or error rate as an evaluation criterion [18,19]. Because wrapper methods search for a feature subset that is best suited to a classifier, they generally obtain better classification performance but at the cost of high time complexity [19]. Embedded methods are special cases of wrapper methods, and feature subsets are obtained when they are used to construct the classifier. This makes them usually more tractable than wrapper methods [20], and there are many embedded methods available and many of them support multiple class problems, such as random forest feature selection, multi-task lasso [21].

Though wrapper methods generally achieve better classification accuracy than filter methods, a major disadvantage is that they are considerably time-consuming. For a dataset with N features, wrapper methods approximately evaluate the quality of $O(N^2)$ feature subsets when using the sequential selection scheme [8], and even incremental wrapper methods handle a linear or sub-quadratic number of candidate feature subsets [22,23]. Such a large number of wrapper evaluations would require a large amount of CPU time when they work on high-dimensional microarray data. To this end, we present a novel model that combines wrapper-based feature selection with the Markov blanket technique. Markov blanket is a cross-entropy based technique that considers the relevance between features, and is capable of explicitly identifying and removing redundant genes. Given the Markov blanket, the eliminated features are conditionally independent of the target class [24], then they have no relevance to the target class, thus can be removed safely. This enables us to identify redundant features in a filter way rather than in a wrapper way and further reduce the number of wrapper evaluations, which leads to better time performance. In addition, it obtains better classification accuracy compared with other methods without introducing Markov blanket, as shown in our preliminary experimental results [25]. The main contributions of this study are as follows. (1) We propose to combine wrapper-based gene selection with the Markov blanket technique to accelerate the feature selection process without degrading the classification performance. Two types of specific feature selectors are implemented based on our approach in this paper. (2) We conducted extensive experiments to verify the effectiveness and efficiency of the proposed methods on ten benchmark microarray datasets with three popular classifiers. The results show our approach outperforms other competing methods. (3) We analyze the theoretical space and time complexity of the proposed approach, and find it is superior in practice. (4) By conducting the feature subset consistency analysis, we find that the resulting set of cancer-predictive genes is not unique. It indicates that there probably exist different subsets of genes in achieving similar or equal predictive classification performance in cancer diagnosis, which facilitates the comprehensive study of disease specific genes.

The rest of this paper is organized as follows. Section 2 briefly illustrates the wrapper-based feature selection methods, symmetric uncertainty, as well as the relevance criteria for feature inclusion. In Section 3, we first introduce several definitions and the Markov blanket, and then detail the proposed feature selection methods. Experimental setting and results are illustrated in Section 4, and Section 5 analyzes the theoretical space and time complexity. Finally, we conclude it with a brief summary.

2. Wrapper-based feature selection

2.1. Wrapper-based feature selection with sequential forward selection

Because wrapper methods use a classifier to measure the quality of a feature subset, they generally obtain low classification error rates due to the specific interaction between the classifier and training set. Obviously, enumerating all combinations of features and evaluating their qualities in turn guarantee obtaining the globally optimal one, but at the cost of high computational complexity that grows exponentially with the number of features [18]. In practice, such high time complexity is often unacceptable, particularly for the gene expression profiles with high dimensionality. To accelerate this process, researchers have proposed various search strategies to generate candidates. In feature selection, commonly used search schemes include, but not limited to, sequential forward selection (SFS), sequential backward selection (SBS), sequential floating search, bidirectional search, random search, and heuristic search [18]. Among these search strategies, SFS achieves a better tradeoff between the quality of the obtained feature subset and the computational complexity. Specifically, initializing the selected feature subset to be empty, SFS selects the first feature that is most relevant to the target class, and then searches for the next candidate feature that most reduces the classification error rate. Continue with the procedure until there is no candidate feature left or no further improvement in classification performance. If k features are finally selected from the total N features, wrapper methods with SFS approximately evaluate $O(kN)$ candidate feature subsets. Algorithm 1 presents corresponding pseudo-code. The *evaluate*() subroutine (Line 7) is the evaluation process for measuring the quality of a candidate gene. The criteria used to select a candidate feature and the notations used in Algorithm 1 are given in subSection 2.2.

2.2. Relevance criteria

In wrapper-based feature selection, the criterion to add a candidate feature f into the selected feature set S is to conduct an inner five-fold cross-validation on training set *Data* projected over $\{S, f\}$ and class label C of *Data*. We use the symbol "↓" to represent the projection over a dataset. For example, $Data^{\downarrow S}$ indicates that we obtain a new dataset that consists of $|S|$ column vectors (selected according to S) from *Data*, i.e., the new dataset is a slice of *Data*. Rather than use the average accuracy of the five-fold cross validation and do a t-test over the cross-validation results [22,29], we adopt the following criteria: (1) a five-fold cross-validation is used on *Data*; (2) the new feature f is selected only if the average accuracy of the five-fold cross-validation over $Data^{\downarrow\{S\cup f\cup C\}}$ is higher than that of the five-fold cross-validation on $Data^{\downarrow\{S\cup C\}}$, and at least MinFoldersBetter (mf) out of the five accuracies over $Data^{\downarrow\{S\cup f\cup C\}}$ is not lower than the average accuracy over $Data^{\downarrow S\cup}$. Such a strategy avoids the criticism for the use of a statistical test on a dataset of small size. Notably, mf is a user-specified threshold. For the better control of low-confidence and over-fitting issues, recommended empirical values for mf are 2 or 3 [8]. The quality of a candidate feature is measured by *evaluate*(*classifier*, $Data^{\downarrow\{S_{new}\cup C\}}$), which returns two items: the average accuracy acc_{new} of the five-fold cross-validation and the number num representing how many times the five accuracies obtained from the five-fold cross-validation over $Data^{\downarrow\{S\cup f\cup C\}}$ are better than average accuracy over $Data^{\downarrow\{S\cup C\}}$.

Algorithm 1. Wrapper-based Sequential Forward Selection (SFS).

```
Input:  Training set Data with feature set F and class label C, mf;
Output: S; //selected features
1   acc = 0;
2   S = {};
3   while NOT isempty(F) do
4       flag = 0;
5       for i = 1 to length(F) do
6           S_new = S ∪ F_i;
7           [acc_new, num] = evaluate(classifier, Data^{↓S_new ∪C});
8           if (acc_new > acc && num ≥ mf) then
9               ind = i;
10              acc = acc_new;
11              flag = 1;
12      if flag then
13          S = S ∪ F_ind; //add it to S
14          F = F - F_ind; //remove it from F
15      else
16          break; //stop feature selection
17  return S;
```

2.3. Incremental wrapper-based subset selection framework

To utilize the advantages of wrapper and filter methods, researchers have proposed to combine them together to achieve a tradeoff between the computational complexity and classification performance [22,23]. Among the various hybrid methods, incremental wrapper subset selection (IWSS) method not only reduces the times of running wrapper evaluations, but also achieves high-quality feature subsets when compared with other state-of-the-art feature selectors [22]. The main idea of IWSS is to use the ranked features, which is obtained by a filter method, to guide the wrapper methods. Algorithm 2 presents the framework of IWSS, which mainly consists of two parts: (1) a filter step that ranks the original features in a descending order using a filter method (lines 1–2); (2) a wrapper step that evaluates the ranked features sequentially (lines 3–8) [23]. Essentially, the *fitness*() subroutine is used to evaluate the quality of a candidate gene f, and it returns one term *select_flag* (a Boolean variable) that indicates whether f is of good quality. We add f to the currently selected feature subset S, if *select_flag* equals one.

Algorithm 2. Incremental wrapper-based subset selection framework.

```
Input:  Training set Data with feature set F and class label C, mf;
Output: S; //selected features
        //filter step
1   R = {}; //feature ranking
2   R = rank(filter, F); //rank F with a filter method
        //wrapper step
3   S = {};
4   for i = 1 to length(R) do
5       S_new = S ∪ R_i;
6       select_flag = fitness(classifier, Data^{↓S_new ∪C});
7       if select_flag then
8           S = S ∪ R_i; //add it to S;
9   return S;
```

2.4. Symmetric uncertainty based feature ranking

According to the framework of IWSS, the first step of IWSS is to rank the original features using a filter method. There are a variety of filter methods available, whereas information theoretic criteria based feature selectors have attracted researchers from many areas and been successfully applied in many fields due to their effectiveness in reflecting the non-linear relationship among variables [17]. In terms of information theory, mutual information is widely used to measure how much the distribution of a predictive variable and the response variable differs from the statistical independence through their non-linear correlation estimation [26]. The more relevant variable shares more information with the target class, which provides us a criterion to rank the predictive variables [17,26]. Due to the fact that the mutual information-based feature ranking method favors a variable with more values, Symmetric Uncertainty (*SU*), i.e. the normalized mutual information, is commonly utilized [27]. *SU* measures the relevance between two variables f and C using the following formula [28]:

$$SU(f,C)=\frac{2*MI(f,C)}{H(C)+H(f)}=\frac{2*(H(C)-H(C|f))}{H(C)+H(f)} \quad (1)$$

where $H(f)$ represents the entropy of f, $MI(f, C)$ is the mutual information between f and C, and $H(C|f)$ measures the conditional entropy quantifying the remaining uncertainty of C given the knowledge of f. Symmetric uncertainty normalizes the value of $MI(f, C)$ to the range [0,1]. A value 0 of $SU(f, C)$ indicates that f and C are completely independent, and a value 1 means that the knowledge of one variable can predict the value of the other. Also, the greater $SU(f, C)$ is, the more relevant f to C. When embedded into the IWSS framework, *SU* ranks the original features in a descending order. IWSS then works over the ranked features sequentially to evaluate the quality of these features. Algorithm 3 outlines the pseudo-code of IWSS using *SU*, where lines 1–4 correspond to the filter step, lines 5–12 illustrate the wrapper step, and lines 9–10 are the metrics for feature inclusion or exclusion. Specifically, Algorithm 3 is a special case of Algorithm 2. In Algorithm 3, we use the symmetric uncertainty metric (a filter method) to rank the genes, and use the *evaluate*() subroutine (illustrated in Algorithm 1 and Section 2.2 relevance criteria) as the *fitness*() function to evaluate the quality of a candidate gene.

Algorithm 3. Incremental Wrapper-based Subset Selection (IWSS).

```
Input:  Training set Data with feature set F and class label C, mf;
Output: S; //selected features
1   R = {}; // ranked features
2   for each f_i in F do
3       su = SU(f_i, C);
4       insert f_i into R in decreasing su value;
5   S = {};
6   acc = 0;
7   for i = 1 to length(R) do
8       S_new = S ∪ R_i;
9       [acc_new, num] = evaluate(classifier, Data^{↓S_new ∪C});
10      if (acc_new > acc && num ≥ mf) then
11          S = S ∪ R_i;
12          acc = acc_new;
13  return S;
```

3. Wrapper-based gene selection with Markov blanket

For wrapper-based feature selection with sequential forward search and incremental selection, they both adopt a greedy forward selection scheme to select candidate features and evaluate the goodness of a candidate feature using a classifier. Though wrapper-based feature selection methods achieve satisfactory classification performance in actual use compared with other well performing feature selectors [22], they are considerably time-consuming [8]. One obvious characteristic

of such methods is that they evaluate each feature using a classifier, even if a candidate feature is redundant to the already selected features. Such a situation could cause wrapper-based feature selection methods a large number of wrapper evaluations and contribute to high time complexity. Assume that a feature f in the candidate feature subset CS is redundant to the selected features S. In wrapper-based feature selection with sequential forward selection (SFS), f is evaluated by the classifier during each iteration and remains in CS until the end of the feature selection. For IWSS, if a low-ranked feature f is redundant to S, f remains in the CS until it is evaluated by the classifier. Obviously, wrapper-based SFS and IWSS can identify redundant features using the classification accuracy, whereas neither of them considers eliminating the redundant features from the candidate feature subset. To alleviate this problem, we propose to integrate the Markov blanket technique into wrapper methods to reduce the times of running wrapper evaluations that are mostly carried out on the redundant features. Specifically, each time when a new feature f is selected into S, the remaining candidate features, whose Markov blanket is f, are to be eliminated from CS, and they are not evaluated by the classifier. Because the eliminated candidate features are conditionally independent of the target class given their Markov blanket, they can be removed safely. Therefore, the proposed approach is expected to improve the feature selection.

3.1. Markov blanket

We first introduce the following notation to help illustrate the proposed approach: *Data* is the samples with n features and m instances and one target variable C; $F=\{F_1, F_2,..., F_n\}$ is the feature space of *Data* and $R=\{R_1, R_2,..., R_n\}$ is a ranking of features in F; $S=\{S_1, S_2,..., S_s\}$ $(1\le s\le n)$ is the currently selected features; F_i, R_i, S_i are the ith feature in F, R and S, respectively.

Definition 1. (*Markov blanket*). Given a feature $F_i\in F$, let $M_i\subset F(F_i\notin M_i)$, M_i is a said to be a Markov blanket of F_i if F_i is conditionally independent of $\{F-M_i-F_i, C\}$ given M_i, i.e.,

$$P(F-M_i-F_i, C)F_i, M_i) = P(F-M_i-F_i, C)M_i) \quad (2)$$

Definition 2. (*Redundant Feature*). Let S be a set of features, a feature f in S is redundant if and only if it has a Markov blanket within S.

In feature selection, if F_i has a Markov blanket M_i within S, it suggests that F_i contributes no more information beyond M_i to the target class, therefore, F_i can be removed safely [24]. Because of the high computational complexity in measuring the conditional independence of features, Yu and Liu proposed to approximate the Markov blanket of F_i using only one feature [27]. Next, we give the definitions of the relevance and redundancy between two variables on the basis of the symmetric uncertainty.

Definition 3. (*C-Relevance*). Given a predictive feature F_i and the target class C, the relevance between them is referred to as *C-Relevance*, noted by $SU(F_i, C)$.

Definition 4. (*F-Relevance*). The correlation between two predictive features F_i and F_j $(i\ne j)$ is referred to as *F-Relevance*, and written as $SU(F_i, F_j)$.

According to the previous discussions, feature selection is actually a process of identifying features with high *C-Relevance* and eliminating *F-Relevance* features. A feature with a larger *C-Relevance* contains more information about the target class than a feature with a smaller *C-Relevance*. To determine the existence of Markov blanket, we can use $SU(F_i, F_j) > SU(F_j,C)$ to test whether F_j contributes to extra information about the class beyond the information from F_i in the case when F_i has a larger *C-Relevance* than that of F_j. An approximate Markov blanket is defined below.

Definition 5. (*Approximate Markov blanket*). Given two predictive features F_i and F_j and the target variable C, if $SU(F_i,C)\ge SU(F_j,C)$ and $SU(F_i, F_j) > SU(F_j,C)$ is satisfied, then F_j is redundant to F_i. F_i is said to be an Approximate Markov blanket of F_j.

3.2. Wrapper-based sequential forward selection with Markov blanket

In the above section, we have seen that Markov blanket has the capacity to decide whether a candidate feature is redundant to the selected features. Considering this, we propose to integrate Markov blanket with wrapper-based sequential forward selection method (SFS-MB). SFS-MB also adopts the greedy forward selection scheme for feature selection, as shown in Algorithm 4. Specifically, SFS-MB first selects the feature from the candidate features that is most relevant to the target class (lines 5–11). Once a feature is selected, SFS-MB adds it to the selected subset S and deletes it from F (lines 13–14). In contrast to the wrapper-based SFS in Algorithm 1, SFS-MB further conducts the redundant feature elimination step, which removes the features, whose Markov blanket is the newly selected one, from the candidate features (lines 15–19). SFS-MB then searches for the next candidate feature that contributes most to the reduction of classification error rate and eliminates features that are redundant to the newly selected one. Repeat the above process until the candidate feature set is empty (line 3) or there is no improvement in classification accuracy (line 21). Because SFS-MB reduces the number of candidate features to be evaluated, it can gain better time performance in comparison with SFS if there exists any conditional independence among the features. Specifically, given a selected feature subset S with $|S|$ features, to determine whether a candidate feature is redundant, the time cost is $O(|S|^2)$ if all selected features need to be compared to. This corresponds to the worst case. Extremely, the candidate feature may be redundant to the first gene of S, then the time complexity is $O(|S|)$.

3.3. Incremental wrapper-based subset selection with Markov blanket

Similar to SFS-MB, we integrate the Markov blanket technique with incremental wrapper-based subset selection (IWSS-MB). Algorithm 5 shows the pseudo-code of IWSS-MB, which comprises two steps. In the first step, all features are ranked in descending order of relevance according to an evaluation metric (lines 1–4). In our study, we use symmetric uncertainty to measure the relevance $SU(f_i, C)$ between each feature f and the target class C. In the second step, a classifier runs over the ranked feature set R sequentially to determine whether a candidate feature should be added into the selected feature subset S (lines 7–17). Specifically, we first evaluate the first ranked feature and include it into S (lines 11–12), because IWSS ensures to select the first ranked feature in R. Rather than directly measure the quality of the second feature using the classifier, IWSS-MB then finds these candidate features whose Markov blanket is the first feature and eliminates them from the candidate features (lines 13–17). Then, IWSS-MB considers the next candidate feature f. If f together with S improves the classification accuracy, IWSS-MB selects f into S and removes the candidates whose Markov blanket is f; otherwise, IWSS-MB considers next candidate feature. Repeat the above process until the last feature in R is evaluated.

Algorithm 4. Wrapper-based Sequential Forward Selection with Markov Blanket (SFS-MB).

```
Input:  Training set Data with feature set F and class label C, mf;
Output: S; //selected features
acc = 0;
S = {};
while NOT isempty(F) do
    flag = 0;
    for i = 1 to length(F) do
        S_new = S ∪ F_i;
        [acc_new, num] = evaluate(classifier, Data^{↓S_new ∪C});
        if (acc_new > acc && num ≥ mf) then
            ind = i;
            acc = acc_new;
            flag = 1;
    if flag then
        S = S ∪ F_ind; //add it to S
        F = F - F_ind; //remove it from F
        //eliminate features redundant to F_ind
        index = {};
        for j = 1 to length(F) do
            if (SU(F_ind,C) ≥ SU(F_j,C)) && (SU(F_ind, F_j)>SU(F_j, C)) then
                index = index ∪ j; //indices of the redundant features
        F = F - F(index);
    else
        break; //stop feature selection
return S;
```

Algorithm 5. Incremental Wrapper-based Subset Selection with Markov Blanket (IWSS-MB).

```
Input:  Training set Data with feature set F and class label C, mf;
Output: S; //selected feature subset
R = {};
for each feature f_i in F do
    Score = SU(f_i , C);
    insert f_i into R in decreasing Score value;
S = {};
acc = 0;
for i = 1 to length(R) do
    S_new = S ∪ R_i;
    [acc_new, num]=evaluate(classifier, Data^{↓S_new ∪C});
    if (acc_new > acc && num ≥ mf) then
        S = S ∪ R_i; //add R_i to S
        acc = acc_new;
        //redundant feature elimination
        f_j = getNextElement(R, R_i);
        while (f_j != null) do
            if SU(R_i, f_j) > SU(f_j, C) do
                R = R - f_j; //remove it from R
            f_j = getNextElement(R, f_j);
return S;
```

4. Experiments and results

4.1. Experimental data

For the purpose of this study as we stated in the introduction section, we conduct extensive experiments on the following ten publicly available microarray datasets that have high-dimensionality and small sample sizes. A brief summary to the ten datasets is presented in Table 1. The last column SFR denotes the ratio between the number of samples and the number of genes and shows a great imbalance between them.

4.1.1. Colon *data*

Colon data consists of 62 samples, and each sample has 2000 genes. Of these samples, 40 are labels as tumor, and the remaining 22 are from normal tissues. The task on this dataset is to distinguish between normal and tumor samples according to the gene expression profiles [30].

Table 1
Experimental dataset description.

ID	Dataset	#Genes	#Samples	#Classes	#SFR
1	*Colon*	2000	62 (40/22)	2	0.031
2	*SRBCT*	2308	83(29/25/11/18)	4	0.036
3	*Leukemia*1	7129	72 (47/25)	2	0.010
4	*Leukemia*2	5327	72(38/9/25)	3	0.014
5	*DLBCL*	7129	77 (58/19)	2	0.011
6	*Prostate*	12600	102(50/52)	2	0.008
7	*Bladder*	5724	40(10/19/11)	3	0.007
8	*Gastric*	4522	30(8/22)	2	0.007
9	*Tox*	5748	171(45/45/39/42)	4	0.030
10	*Blastomi*	1465	23(10/13)	2	0.016

4.1.2. *Small Round Blue Cell Tumor (*SRBCT*) data*

SRBCT has four different types of childhood tumors: Ewing's family of tumors (EWS), neuroblastoma (NB), non-Hodgkin Burkitt's lymphoma (BL), and rhabdomyosarcoma (RMS). There are 83 samples in *SRBCT* and each sample contains 2308 genes. Of these samples, 29 are EWS samples, 18 are NB samples, 11 BL and 25 RMS samples. The goal is to build a classifier to distinguish the four subtypes of tumors [31].

4.1.3. Leukemia*1 data*

This dataset contains 72 samples that are collected from bone marrow and peripheral blood in leukemia patients to distinguish between acute lymphoma leukemia (ALL) and acute myeloid leukemia (AML) tissues. Of these samples, 25 are AML samples, 47 are ALL samples and there are 7129 genes in each sample [2]. The classification task on *Leukemia*1 is to distinguish these two types of leukemia.

4.1.4. Leukemia*2 data*

A collection of leukemia patient samples from bone marrow and peripheral blood is used for distinguishing between acute myeloid leukemia (AML) and acute lymphoma leukemia (ALL) tissues. The data for ALL samples are further divided into B-cell ALL and T-cell ALL. *Leukemia*2 consists of 72 samples with 5327 genes: 25 AML samples, 38 ALL-B samples and 9 ALL-T samples [2]. The classification goal is to classify the three subtypes of leukemia.

4.1.5. *Diffuse Large-B-Cell Lymphoma (*DLBCL*) data*

This dataset is a collection of B-cell lineage malignancy diffuse large B-cell lymphomas (BCL) and follicular lymphomas (FL) samples. *DLBCL* consists of 19 FL samples and 58 BCL samples. Each sample is described by 7129 genes [32].

4.1.6. Prostate *data*

This dataset consists of 50 normal samples and 52 prostate tumors. The number of genes is 12600 [33]. The task is to identify the gene expression patterns that distinguish the tumor from normal.

4.1.7. Bladder *data*

This dataset is a collection of bladder carcinoma samples from three different tumor stages. There are 10 samples in tumor stage T2-T4, 19 samples in stage Ta and 11 samples in stage T1. Each sample contains 5724 genes [34]. The task is to build a classifier to classify the bladder tumor samples.

4.1.8. Gastric *data*

Gastric consists of tumor gastric samples and normal gastric samples. There are 30 samples in total, and each sample is descried

by 4522 genes. Of these samples, 8 samples are from the normal gastric tissue and 22 samples are from the tumor gastric tissues [35]. The task is to induce a classification model to distinguish the tumor gastric samples from the normal.

4.1.9. Tox *data*

The number of genes in each sample is 5748. *Tox* includes four different subtypes, each with 45, 45, 39, and 42 samples, respectively. The classification task is to classify the four subtypes of *Tox* [36].

4.1.10. Blastomi *data*

it consists of 10 examples from the metastatic medulloblastoma and 13 examples from the non-metastatic medulloblastoma. Each sample is described by 1465 genes. The task on this dataset is to build a prediction model to distinguish metastatic medulloblastoma from non-metastatic medulloblastoma [37].

4.2. Experimental setup

In wrapper-based feature selection, a classifier is required to be used as the evaluation function to measure the goodness of candidate features. In addition, to verify the effectiveness of wrapper methods in feature selection, a classifier is also required to evaluate the quality of the finally obtained feature subset. In our study, three commonly used classifiers with different metrics are used, including *k*-nearest-neighbor, naïve Bayes and C4.5 decision tree. Moreover, the same classifier is not only integrated into the wrapper procedure to evaluate the goodness of a candidate feature, but also used as the classifier to evaluate the quality of finally obtained feature subset. For microarray data characterized by high dimensionality and small sample sizes, to evaluate the quality of the obtained features, a ten-fold cross-validation is favorable for generating independent training set and test set [38], and in this process, each one of the ten folds is retained as a test set to evaluate the quality of the finally obtained features whereas the remaining nine folds are used as the training set. In particular, feature selection is performed only on the training set for an unbiased selection protocol [39,40]. The classifier is then trained on the training set projected over the selected features and evaluated on the test set projected over the selected features. The final accuracy is the mean of the ten classification results. Furthermore, for the easy calculation of symmetric uncertainty in handling continuous variables, we first normalize each gene with zero mean and one standard deviation, and then discretize the continuous values into three disjointed partitions with two thresholds −0.5 and 0.5 [41].

Because the goal of our experiments is to test the effectiveness of integrating the Markov blanket technique into wrapper-based feature selection, we focus our experiments on comparison between wrapper-based feature selection with Markov blanket and methods without Markov blanket. To demonstrate the effectiveness and efficiency of the proposed approach, experiments are conducted in terms of the accuracy, the size of finally obtained features, the number of wrapper evaluations and the actual time costs. As for the effectiveness of wrapper-based feature selection without Markov blanket, please refer to prior work [22,23] for details on its performance in comparison with other well-performing feature selectors (e.g. correlation-based feature selection, fast correlation based filter, and FOCUS). We further use a Wilcoxon signed-rank test with a significance interval of 95% to determine whether there is any difference between wrapper-based feature selection without and with Markov blanket in classification accuracy and the size of obtained feature subset [42]. The difference is significant if the *p*-value is less than 0.05. We implemented these algorithms with Matlab and ran experiments on a Quad-core Intel CPU (with a 3.2 GHz processor and 4 G RAM).

Table 2
Experimental results of SFS and SFS-MB in accuracy and the size of selected genes.

ID	SFS^2		SFS^2-MB		SFS^3		SFS^3-MB	
	acc	gene	acc	gene	acc	gene	acc	gene
1	72.6	4.0	**77.4**	3.3	76.0	3.1	76.7	3.2
2	86.8	6.6	78.5	6.4	**89.4**	5.7	86.2	6.2
3	87.2	2.0	84.3	2.0	88.6	2.1	**89.9**	2.2
4	83.0	3.4	**92.1**	3.6	84.9	4.1	82.0	3.6
5	80.9	3.8	**87.3**	4.0	79.4	3.3	84.6	3.1
6	79.5	3.4	**80.4**	3.6	**80.4**	4.7	77.5	4.0
7	57.5	3.0	73.3	3.0	69.7	2.9	**74.0**	2.7
8	97.5	1.0	**100.0**	1.0	96.7	1.0	96.7	1.0
9	55.9	7.2	58.9	8.0	59.2	6.9	**64.7**	7.4
10	51.7	2.2	**66.7**	1.5	46.7	1.8	55.0	1.3
Test			=	=			=	=
1NN								
1	79.0	5.3	77.6	3.8	77.4	4.3	**82.4**	3.7
2	82.7	5.7	**87.6**	6.1	84.0	6.8	79.8	5.4
3	91.4	2.8	**91.5**	2.8	83.8	2.8	88.9	2.8
4	**95.5**	3.5	94.3	3.6	90.2	3.5	86.1	3.5
5	87.0	3.8	85.4	4.1	**87.5**	4.2	78.7	3.6
6	81.5	5.1	84.5	4.4	81.3	4.7	**88.5**	4.2
7	70.0	3.1	78.5	3.2	64.3	3.6	74.2	3.5
8	93.3	1.0	**100.0**	1.0	88.3	1.0	90.0	1.0
9	66.2	7.7	68.0	8.3	68.0	8.8	**68.7**	9.1
10	58.3	2.4	66.7	2.4	**68.3**	3.0	**68.3**	3.1
Test			=	=			=	=
NB								
1	**79.0**	3.6	76.0	3.8	74.5	4.7	77.4	2.8
2	**85.9**	4.7	83.1	4.5	85.1	4.9	84.7	4.8
3	81.7	2.6	78.6	2.2	83.9	2.5	**84.3**	2.8
4	86.1	3.4	76.5	3.3	77.7	3.4	**87.7**	3.6
5	81.8	3.3	80.7	3.5	77.5	2.9	**84.8**	3.2
6	**88.4**	4.1	86.5	4.3	78.6	4.1	85.3	3.9
7	72.0	3.4	71.3	2.8	69.7	3.4	**76.8**	3.2
8	97.5	1.0	**100.0**	1.0	97.5	1.0	**100.0**	1.0
9	52.4	4.8	**59.1**	4.6	51.6	5.0	52.6	5.5
10	56.7	2.3	**63.3**	2.2	41.7	1.8	55.0	2.4
Test			=	=			*	=
C4.5								

4.3. Experimental results of wrapper-based sequential forward selection

In this section, we present the experimental results of wrapper-based sequential forward selection (SFS) and wrapper-based sequential forward selection with Markov blanket (SFS-MB), and consider both the cases where *mf* ={2, 3}, marked as the superscript on each method. Table 2 presents the experimental results for 1-nearest-neighbor (1NN), naïve Bayes (NB) and C4.5, respectively. For each dataset, the average of the ten-fold classification accuracy, and the average of the size of the ten selected features subsets are given. The best accuracy achieved on each microarray dataset is shown in bold. The last row "Test" shows the results of the Wilcoxon signed-rank test with a confidence level of α=0.05 between SFS and SFS-MB. A notation "*" in the tables represents that SFS-MB achieves better accuracy or a smaller subset of features in comparison with the one without Markov blanket, "^" represents that SFS-MB obtains worse accuracy or a greater subset of features, and "=" means that there is no statistical difference between SFS and SFS-MB.

For 1NN, regarding accuracy, we can observe that SFS-MB achieves 79.9% classification accuracy averaged over the ten datasets, which is comparable to 75.3% obtained by SFS in the case of *mf* =2. When *mf* =3, SFS-MB achieves 78.7% average classification accuracy and SFS achieves 77.1% classification accuracy. In both cases, there is no statistically significant difference between SFS and SFS-MB. Regarding the size of finally obtained feature subset, the average size

selected by SFS-MB is 3.6 for *mf* =2 and 3.5 for *mf* =3, while SFS selects 3.7 features for *mf* =2 and 3.6 features for *mf* =3. In both cases, there is no significant difference between SFS and SFS-MB. Further, we can observe that SFS with Markov blanket outperforms SFS on nine out of the ten experimental datasets, except for the *SRBCT* dataset. On *SRBCT*, the best accuracy achieved by SFS is 89.4%, which is comparable to 86.2% of SFS-MB. In this case, SFS-MB with *mf* =2 tend to be a good choice. Similar conclusions can be drawn for NB and C4.5. Specifically, for NB, SFS^2 achieves 80.5% average accuracy, and SFS^2-MB achieves 83.4% average accuracy. When *mf* =3, SFS achieves 79.3% average accuracy and SFS-MB 80.6% average accuracy. The Wilcoxon signed-rank test shows that there is no significant difference between them. In this case, SFS with Markov blanket performs better than SFS on eight out of the ten experimental datasets, where SFS^2-MB achieves the best accuracy three times and SFS^3-MB obtains the best accuracy on four datasets. SFS^3-MB coupled with NB seems to a good combination. For C4.5 as shown, we can see that there is no significant difference between SFS and SFS-MB in the size of the final obtained feature subset. Notably, SFS^2-MB achieves comparable classification accuracy to SFS^2, and SFS^3-MB achieves better accuracy than SFS^3 with a *p*-value of the Wilcoxon signed-rank test less than 0.05. In this situation, SFS without Markov blanket obtains the best accuracy on three datasets, while SFS with Markov blanket outperforms SFS on seven datasets. Typically, SFS^3-MB achieves the best accuracy on five out of the seven datasets. Overall, the experimental results presented in Table 2 indicate that the proposed SFS with Markov blanket obtains comparable accuracy to or better accuracy than SFS without Markov blanket.

In the previous section, it is concluded that SFS-MB achieves comparable performance to SFS in terms of the classification accuracy and the size of finally obtained feature subset. In this section, we investigate the number of wrapper evaluations carried out in each approach. Table 3 presents experimental results using 1NN, NB and C4.5, respectively. The last row "Ratio" is the ratio between the average number of wrapper evaluations of SFS and the average number of wrapper evaluations of SFS-MB. According to results in Table 3, we can observe that compared to the situation without Markov blanket, SFS-MB greatly reduces the number of wrapper evaluations conducted in feature selection on each dataset. For instance, when using 1NN, for *mf* =2, the average number of wrapper evaluations of SFS is 24362.0, whereas the average number of wrapper evaluations of SFS-MB is 10922.0. For *mf* =3, SFS-MB reduces the average number of wrapper evaluations from 25437.0 to 10819.0. For NB, the ratio between SFS^2 (SFS^3) and SFS^2-MB (SFS^3-MB) is 2.6 (2.6), and for C4.5, the ratio between SFS and SFS-MB is 2.3.

Besides presenting the number of wrapper evaluations, to verify the performance gain of SFS with Markov blanket in time cost, we present the ratio between the time costs of SFS and time costs of SFS-MB for each dataset in Fig. 1. For each dataset, we present the time costs associated with 1NN, NB, and C4.5, respectively. The X-axis indicates different classifiers, and Y-axis, named "Speedup", represents the speedup ratio. A value of "Speedup" larger than one means that SFS with Markov blanket achieves better time performance compared with SFS without Markov blanket. From Fig. 1, we observe that SFS with Markov blanket accelerates the feature selection process impressively compared to the one without Markov blanket. For 1NN, for instance, on *Colon*, SFS^2 costs 667.6 s, while SFS^2-MB costs 195.1 s, and SFS^3 costs 547.4 s, while SFS^3-MB costs 210.4 s. On *Prostate*, SFS^2 costs 3880.0 s and SFS^3 costs 5037.1 s, while SFS^2-MB only costs 1234.1 s and SFS^3-MB costs 1250.9 s. For NB, typically on *Prostate*, SFS^2 costs 6074.4 s and SFS^3 costs 5662.1 s, while SFS^2-MB costs 1389.0 s and SFS^3-MB costs 1487.6 s. For C4.5, on *Prostate*, SFS^2-MB (SFS^3-MB) costs 12043.0 (11663.0) seconds compared to 42367.0 (42349.0) seconds of SFS^2 (SFS^3).

Table 3
Number of required wrapper evaluations for SFS and SFS-MB.

ID	SFS^2	SFS^2-MB	SFS^3	SFS^3-MB
1	9989.2	2876.8	8193.2	3097.3
2	15462.0	9099.0	13634.0	8931.7
3	21384.0	11433.0	22097.0	11619.0
4	23431.0	13155.0	27157.0	13146.0
5	34210.0	12340.0	30647.0	11660.0
6	55432.0	17402.0	71806.0	17633.0
7	22890.0	12551.0	22318.0	12010.0
8	9043.0	6404.4	9043.0	6402.8
9	47095.0	21938.0	45379.0	21717.0
10	4684.3	2016.0	4099.2	1968.7
Ratio		2.2		2.4
1NN				
1	12581.0	2987.3	10588.0	3144.7
2	13635.0	8832.1	15869.0	8260.6
3	27085.0	12298.0	27085.0	12390.0
4	23964.0	13040.0	23963.0	12954.0
5	34210.0	11742.0	37059.0	11473.0
6	76844.0	17479.0	71806.0	18049.0
7	23462.0	13345.0	26322.0	13409.0
8	9043.0	6258.3	9043.0	6287.9
9	49973.0	22948.0	56285.9	23514.4
10	4976.8	2117.6	5853.7	2289.1
Ratio		2.5		2.5
NB				
1	9191.3	2987.4	11386.0	2738.9
2	11603.0	7676.1	12009.0	7691.7
3	25660.0	11525.0	24947.0	12938.0
4	23431.0	12949.0	23431.0	13357.0
5	30648.0	12425.0	27797.0	11645.0
6	64249.0	17433.0	64249.0	16850.0
7	25178.0	13297.0	25178.0	14353.0
8	9043.0	6410.9	9043.0	6409.0
9	33324.0	17145.0	34470.0	18769.0
10	4830.6	2146.7	4099.4	2269.0
Ratio		2.3		2.2
C4.5				

4.4. Experimental results of incremental wrapper-based subset selection

In this section, we present the experimental results of incremental wrapper-based feature subset selection without (IWSS) and with Markov blanket (IWSS-MB), and consider both the cases where *mf* ={2, 3}.

Table 4 presents the results for 1NN, NB and C4.5. For each dataset, the average of the ten-fold classification accuracy, and the average of the size of the ten selected features are given. The best accuracy achieved on each microarray dataset is shown in bold. The last row "Test" shows the results of the Wilcoxon signed-rank test with a confidence level of α=0.05 between the approach with and without Markov blanket. A notation "*" in the tables represents that Markov blanket embedded method achieves better accuracy or a smaller subset of features in comparison with the method without Markov blanket, "^" represents that Markov blanket embedded method obtains worse accuracy or a greater subset of features, and "=" means that there is no statistical difference between the two approaches.

As shown in Table 4 for 1NN, regarding accuracy, we can observe that IWSS-MB achieves 86.2% classification accuracy averaged over the ten datasets, which is comparable to 82.8% obtained by IWSS in the case of *mf* =2, and that $IWSS^3$-MB achieves 83.7% average classification accuracy and $IWSS^3$ achieves 82.7% average accuracy. In both cases, there is no significant difference between IWSS and IWSS-MB with a confidence level of α=0.05. Regarding the size of finally obtained

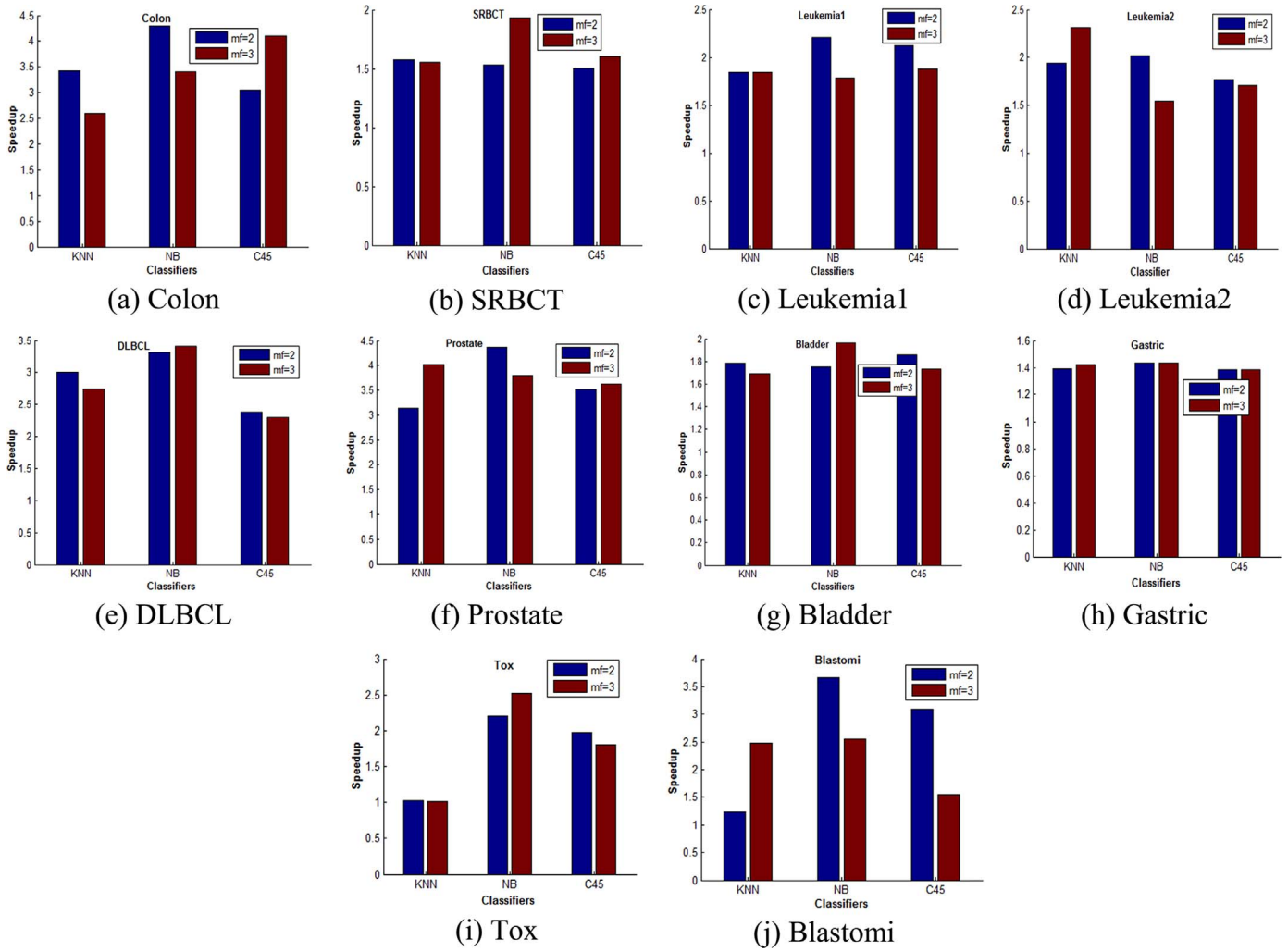

Fig. 1. Running time comparison(s) for SFS and SFS-MB.

feature subset, the average size obtained by IWSS-MB is 8.1 in the case of *mf* =2, which is much smaller than 10.2 of IWSS. In the case of *mf* =3, IWSS-MB selects 6.7 features and IWSS selects 8.0 features. Further, we can observe that IWSS with Markov blanket outperforms IWSS on six out of the ten experimental datasets. On the other four datasets, the classification accuracy obtained by IWSS-MB is comparable to that of IWSS. In this case, IWSS-MB with *mf* =2 tend to be a good choice. For NB, IWSS-MB achieves comparable classification accuracy to IWSS, and when *mf* =2, IWSS-MB selects 7.6 features, which is much smaller than 9.3 features of IWSS. In this case, IWSS with Markov blanket performs better than IWSS on nine out of the ten experimental datasets, where $IWSS^2$-MB achieves the best accuracy on six datasets and $IWSS^3$-MB obtains the best accuracy on three datasets. So, $IWSS^2$-MB coupled with NB seems to a good combination. Also, for C4.5, we observe that there is no significant difference in classification accuracy between IWSS-MB and IWSS, however, IWSS-MB selects a smaller subset of features in comparison with IWSS for both *mf* =2 and *mf* =3. In this situation, IWSS without Markov blanket obtains the best accuracy on three datasets, while IWSS with Markov blanket outperforms IWSS on seven datasets. Typically, $IWSS^3$-MB achieves the best accuracy on five out of the seven datasets. When using C4.5 classifier, we would like to use IWSS with Markov blanket with *mf* =3. Overall, experimental results in Table 4 indicate that the proposed incremental wrapper based subset selection with Markov blanket achieves comparable accuracy to IWSS.

Next, we investigate the number of wrapper evaluations required in IWSS and IWSS-MB, respectively. The last row "Ratio" is the ratio between the average number of wrapper evaluations of IWSS and the average number of wrapper evaluations of IWSS-MB. Table 5 presents the number of wrapper evaluations when using 1NN, NB and C4.5. According to results in Table 5, we can observe that IWSS-MB greatly reduces the number of required wrapper evaluations in comparison with IWSS on all the datasets for each classifier. For instance, the average number of wrapper evaluations of $IWSS^2$ ($IWSS^3$) is 5395.2 (5395.2), whereas the average number of wrapper evaluations of $IWSS^2$-MB ($IWSS^3$-MB) is 479.6 (475.0) when using 1NN; the ration between the average number of wrapper evaluations of $IWSS^2$ ($IWSS^3$) and that of $IWSS^2$-MB ($IWSS^3$-MB) is 12.6 (11.8) when using NB; the average number of wrapper evaluations of $IWSS^2$ ($IWSS^3$) is 11.1 (10.0) times more than that of $IWSS^2$-MB ($IWSS^3$-MB) in the case of C4.5.

We then conduct experiments to show the performance gain in time cost and present the ratio between the time costs of IWSS and time costs of IWSS-MB for each dataset in Fig. 2. For each dataset, we present the time costs associated with 1NN, NB, and C4.5, respectively. The X-axis indicates different classifiers, and Y-axis, named "Speedup", represents the speedup ratio. A value of "Speedup" larger than one means that IWSS with Markov blanket achieves better time performance compared with IWSS without Markov blanket. From Fig. 2, we observe that IWSS with Markov blanket accelerates the feature

Table 4
Experimental results of IWSS and IWSS-MB in accuracy and the size of selected genes.

ID	$IWSS^2$		$IWSS^2$-MB		$IWSS^3$		$IWSS^3$-MB	
	acc	gene	acc	gene	acc	gene	acc	gene
1	76.0	13.5	71.0	5.4	**80.2**	10.0	78.8	4.4
2	92.6	11.8	93.8	11.8	**96.5**	12.7	95.0	11.4
3	**96.1**	8.3	95.7	7.3	92.0	6.6	90.5	7.7
4	96.9	4.3	95.7	4.1	**98.6**	5.2	95.2	5.5
5	85.9	12.4	**93.4**	9.1	92.0	9.9	91.1	9.7
6	88.3	12.0	**90.4**	9.0	85.5	11.2	88.2	6.5
7	65.2	8.8	**81.7**	7.3	74.7	7.7	81.0	7.2
8	96.7	1.1	**100.0**	1.3	**100.0**	1.1	**100.0**	1.1
9	68.7	24.8	**77.0**	22.0	69.0	23.7	69.2	18.9
10	61.7	4.7	**63.3**	3.7	38.3	4.6	48.3	3.7
Test			=	*			=	=
1NN								
1	**85.7**	11.1	79.0	5.3	81.0	10.4	83.6	4.2
2	88.0	12.5	**94.0**	11.8	92.8	10.8	91.5	11.4
3	94.0	7.5	94.6	7.1	93.8	6.7	**94.8**	6.7
4	95.9	6.6	**96.3**	6.2	95.7	6.1	95.5	6.3
5	89.8	15.4	89.3	10.3	88.6	13.5	**90.0**	10.6
6	91.2	13.5	**94.0**	8.2	92.3	11.3	92.3	7.1
7	79.7	9.8	**81.3**	7.8	77.5	8.6	79.0	7.0
8	94.2	1.3	**100.0**	1.5	96.7	1.3	96.7	1.4
9	68.5	31.3	**72.2**	23.8	69.1	24.4	69.5	19.2
10	35.0	7.0	55.0	5.4	60.0	6.4	**66.7**	4.2
Test			=	*			=	=
NB								
1	72.6	8.3	77.4	4.2	**81.9**	8.4	81.0	3.5
2	86.4	11.3	84.5	9.0	78.5	11.1	**86.8**	8.7
3	89.9	6.9	90.5	5.9	91.5	6.2	**92.0**	5.6
4	90.3	6.7	92.6	7.1	**94.0**	7.6	87.1	6.0
5	80.4	11.6	87.0	7.9	85.5	10.3	**87.5**	6.7
6	**88.2**	11.0	86.5	5.7	83.4	10.5	85.5	5.4
7	70.7	10.0	**74.7**	8.4	**74.7**	8.3	74.0	7.5
8	**100.0**	1.2	**100.0**	1.1	**100.0**	1.0	**100.0**	1.0
9	56.3	15.3	**58.9**	13.2	56.7	13.9	58.5	12.3
10	48.3	4.8	55.0	4.2	55.0	4.1	**56.7**	3.1
Test			*	*			=	*
C4.5								

Table 5
Number of required wrapper evaluations for IWSS and IWSS-MB.

ID	$IWSS^2$	$IWSS^2$-MB	$IWSS^3$	$IWSS^3$-MB
1	2000.0	15.2	2000.0	29.5
2	2308.0	481.4	2308.0	517.1
3	7129.0	290.1	7129.0	305.3
4	5327.0	1166.6	5327.0	909.1
5	7129.0	201.3	7129.0	197.4
6	12600.0	78.5	12600.0	104.0
7	5724.0	765.1	5724.0	750.5
8	4522.0	1340.4	4522.0	1345.9
9	5748.0	382.2	5748.0	491.6
10	1465.0	75.4	1465.0	99.7
Ratio		11.2		11.4
1NN				
1	2000.0	18.5	2000.0	28.8
2	2308.0	463.1	2308.0	510.2
3	7129.0	320.5	7129.0	338.9
4	5327.0	818.3	5327.0	805.4
5	7129.0	188.3	7129.0	164.9
6	12600.0	87.1	12600.0	104.2
7	5724.0	597.1	5724.0	681.1
8	4522.0	1209.8	4522.0	1332.5
9	5748.0	384.4	5748.0	463.0
10	1465.0	49.9	1465.0	75.7
Ratio		13.0		12.0
NB				
1	2000.0	28.1	2000.0	31.0
2	2308.0	614.1	2308.0	675.4
3	7129.0	442.1	7129.0	422.6
4	5327.0	768.8	5327.0	894.6
5	7129.0	244.8	7129.0	344.4
6	12600.0	129.9	12600.0	154.8
7	5724.0	618.7	5724.0	659.5
8	4522.0	1361.1	4522.0	1512.5
9	5748.0	676.1	5748.0	710.0
10	1465.0	85.7	1465.0	101.8
Ratio		10.9		9.8
C4.5				

selection process impressively compared to the one without Markov blanket.

Overall, according to the experimental results and analysis from both the wrapper-based feature with sequential forward selection and the incremental wrapper-based feature selection methods, we conclude that in comparison with wrapper methods without Markov blanket, Markov blanket embedded methods obtain comparable classification accuracy, substantially reduce the number of wrapper evaluations, and speed up the feature selection process impressively for both the binary and multiclass classification. This demonstrates the effectiveness and efficiency of our proposed approach.

4.5. Experimental results of feature subset consistency

In this section, to measure the degree of consistency of two feature subsets selected by two different feature selection methods, we present the experimental results of wrapper-based feature selection methods with and without Markov blanket. Table 6 shows the results of SFS vs. SFS-MB, and IWSS vs. IWSS-MB for both the cases where mf ={2, 3}. The experimental results are obtained by running the feature selector over the training set. Each entry in Table 6 represents the number of features selected without using Markov blanket, the number of features selected with Markov blanket, and the number of common features selected by two feature selectors. For instance, the entry "8/3/1" of $IWSS^3$ and $IWSS^3$-MB for 1NN indicates that $IWSS^3$ selects 8 features, $IWSS^3$-MB selects 3 features, and there is one common feature between them.

According to the experimental results in Table 6, we observe that wrapper-based feature selector with Markov blanket selects a smaller number of features in comparison with the case without Markov blanket.

This indirectly demonstrates the efficiency of the proposed method in selecting discriminant features. In addition, we can observe that few common features are selected by two feature selectors, and there is even no common feature selected. For instance, for the case of SFS^2 vs. SFS^2-MB of 1NN, the entry is "5/3/0"; for the case of $IWSS^2$ vs. $IWSS^2$-MB, the entry is "14/5/2". This indicates that there exist different subsets of genes achieving similar or equal cancer-predictive classification performance, which can help researchers obtain several diverse subsets of genes associated with a specific cancer and look deeper into the disease mechanism in a comprehensive way. Specifically, we then take a further step to investigate the gene function of different gene sets obtained by different feature selection methods. Without loss of generality, we take medulloblastoma microarray data as an example. For the case of NB, the subset of genes selected by SFS^2 includes {ACTB, WT1-AS, IL10}, and SFS^2-MB selects {IL10, SMAD5, ACTB, EIF4A1}; {GSTO1, THRA, N4BP2L1, HTRA1} and {YWHAE, CSHL1, MT3} are the subsets of genes obtained by SFS^3 and SFS^3-MB, respectively. For IWSS and IWSS-MB, $IWSS^2$ selects {GAPDH, CXCR5, VIP, FGFR1, TPBG} in comparison to {GAPDH, RAB2A, ACTB, XRCC5} of $IWSS^2$-MB. For mf =3, $IWSS^3$ selects {GAPDH, CXCR5, BLM, VAT1, NTRK1}, and $IWSS^3$-MB obtains {GAPDH, TXK, GSTP1, DDR1, IGFBP2}. We then list the gene function of the two gene subsets obtained by $IWSS^3$ and $IWSS^3$-MB using NB, respectively (shown in

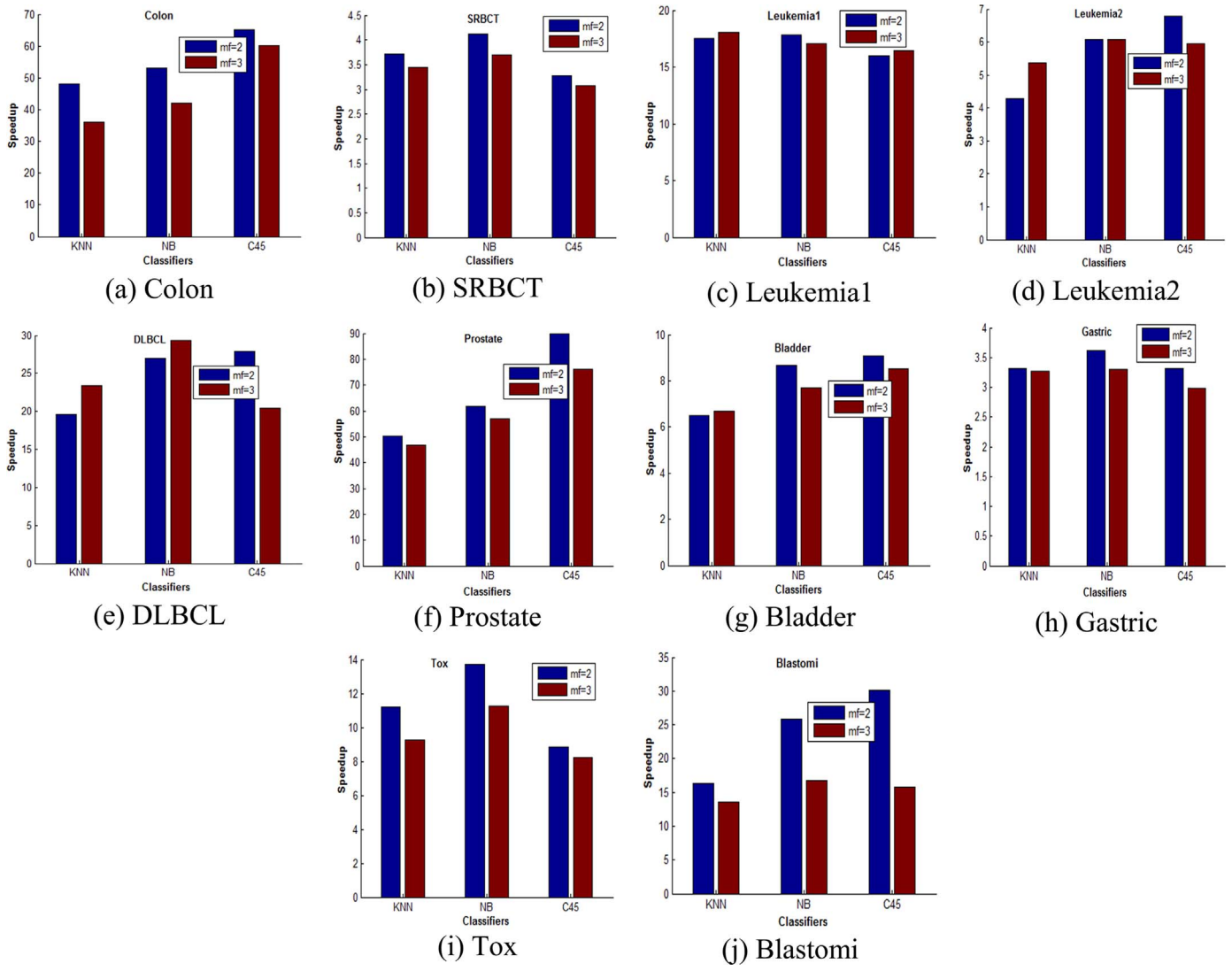

Fig. 2. Running time comparison(s) for IWSS and IWSS-MB.

Table 7). We can observe that both the two gene subsets play an important role in the biological process, though the two gene subsets only share one common gene.

5. Time and space complexity analysis

The above section shows the superiority of our proposed method in reducing the time cost and achieving high classification accuracy for both the binary and multi-class problems. In this section, we analyze the theoretical space and time complexity of the two types of feature selectors: the case of sequential forward selection and the case of incremental wrapper subset selection. For the space complexity analysis, we need to load the training set into memory and store the trained classifier for prediction purpose. In particular, for a training set with m samples and n features, if using the Markov blanket technique, we need to store the relevance values between each feature with the class label (O(n) space complexity), and store the redundancy values between each pair of features (O(n^2) space complexity). Therefore, in comparison with the case without using Markov blanket, we have an extra space complexity of O(n^2+n) in the RAM memory, which is easily affordable in current practices.

For the time complexity analysis, the time complexity of wrapper-based feature selector is mainly determined by the number of required wrapper evaluations and the complexity of the classifier used to evaluate the goodness of a candidate feature subset S. The complexity of a classifier is mainly determined by the training set size m and the size s of S, that is, it is a function of m and s, and we note it as $g(m, s)$. Because m is a constant in feature selection, we simplify it as O($g(s)$). Both average and worst time complexity are analyzed.

5.1. Sequential forward selection case

5.1.1. Without using Markov blanket

SFS needs to conduct wrapper evaluations n times in selecting the first feature, and the classifier complexity is O(g(1)); in selecting the second feature, it conducts wrapper evaluations (n-1) times, and the classifier complexity of O(g(2)). Suppose s features are finally selected, in the last run, SFS conducts (n-s) times wrapper evaluations with a classifier complexity of O(g(s+1)). Thus, we obtain the overall time complexity:

n*O(g(1))+ (n−1)*O(g(2))+…+(n−s)*O(g(s+1))= $\sum_{i=0}^{s}(n-i)\mathrm{O}(g(i+1))$. If all of the features are selected, the time complexity is $\sum_{i=0}^{n}(n-i)\mathrm{O}(g(i+1))$. The best case is that only one feature is selected, i.e., s=1.

5.1.2. Using Markov blanket

Because the number of removed redundant features in each iteration is a dynamic number, we analyze the time complexity of

Table 6
Feature subset consistency comparison.

ID	SFS^2 vs. SFS^2-MB	SFS^3vs. SFS^3-MB	$IWSS^2$ vs. $IWSS^2$-MB	$IWSS^3$ vs. $IWSS^3$-MB
1	5/3/0	5/2/0	14/5/2	8/3/1
2	6/6/0	6/7/0	10/11/5	15/9/5
3	2/2/1	3/2/0	10/7/3	7/8/1
4	4/6/0	4/4/0	4/4/3	3/4/3
5	3/3/0	3/3/1	11/10/5	11/6/3
6	4/4/0	6/5/0	13/8/3	10/8/4
7	3/5/1	3/3/1	9/9/2	5/4/4
8	1/1/1	1/1/1	1/1/1	1/1/1
9	6/14/0	18/11/1	32/18/3	29/22/3
10	1/1/0	1/1/0	3/6/1	3/3/2
1NN				
1	5/2/1	4/2/2	10/4/2	16/4/1
2	6/6/1	9/7/1	11/15/3	11/15/6
3	2/3/1	3/3/0	8/7/5	8/6/3
4	3/3/2	4/4/1	8/4/3	7/6/3
5	6/3/0	5/5/0	13/13/2	13/13/2
6	6/7/1	6/4/0	11/10/2	14/6/2
7	3/4/1	4/3/0	11/10/4	10/8/4
8	1/1/1	1/1/0	1/2/1	1/1/1
9	7/12/0	4/14/1	32/25/4	16/18/3
10	3/4/2	4/3/0	5/4/1	5/5/1
NB				
1	1/4/0	4/4/1	11/4/1	11/3/1
2	4/4/1	5/5/0	17/8/4	12/6/3
3	2/3/1	3/3/0	8/7/5	8/6/3
4	3/3/0	3/3/1	10/4/3	10/7/3
5	3/3/1	3/3/0	15/7/3	9/10/2
6	5/4/0	6/4/2	13/5/2	8/5/2
7	3/4/1	4/3/0	11/10/4	10/8/4
8	1/1/1	1/1/1	1/1/1	1/1/1
9	5/3/1	5/3/1	12/16/1	12/14/2
10	1/2/0	3/2/0	7/5/2	7/4/3
C4.5				

SFS-MB with the following situations. a) If there exists no redundancy among features, SFS-MB then does not eliminate features, and its time complexity is comparable to that of SFS. b) The first selected feature is the Markov blanket of all the left features, these features are then to be removed after selecting the first feature. The time complexity is $O(n*g(1))$. c) Only one feature is removed. Typically, we can assume that the elimination occurs in the process of finding the first feature. Then, SFS-MB conducts n times wrapper evaluations in selecting the first feature with the classifier complexity $O(g(1))$; in obtaining the second feature, it conducts (n-2), rather than (n-1), times wrapper evaluations, and the classifier complexity of $O(g(2))$. Suppose s features are finally selected, in the last run, SFS-MB conducts (n-s-1), rather than (n-s), times wrapper evaluations with a classifier complexity of $O(g(s+1))$. Thus, the overall time complexity is $n*O(g(1))+ (n-2)*O(g(2))+\ldots+(n-s-1)*O(g(s+1))$, which is less than that of SFS. d) More than one features are removed. According to the analysis of case c), we can obtain a lower time complexity.

5.2. Incremental wrapper subset selection case

5.2.1. Without using Markov blanket

IWSS conducts once wrapper evaluation in selecting the first feature, and the classifier complexity is $O(g(1))$; in selecting the second feature, and the classifier complexity of $O(g(2))$. In evaluating the last feature, the classifier complexity of IWSS is $O(g(n))$. Thus, the overall time complexity is: $O(g(1))+O(g(2))+\ldots+O(g(n))=\sum_{i=1}^{n} O(g(i))$.

5.2.2. Using Markov blanket

Similar to SFS, we analyze the time complexity of IWSS-MB in the following four situations. a) If there exists no redundancy among features, IWSS-MB then does not eliminate features, and has a time complexity comparable to that of IWSS. b) The first selected feature is the Markov blanket of all the left features, these features are then to be removed after the first feature is selected. The time complexity is $O(g(1))$. c) Only one feature is removed. Typically, we can assume that the elimination occurs in the process of evaluating the first feature. Then, IWSS-MB has a classifier complexity $O(g(1))$ in selecting the first

Table 7
Selected gene and its function.

Gene	Gene function
GAPDH	It encodes a member of the glyceraldehyde-3-phosphate dehydrogenase protein family. Its product catalyzes an important energy-yielding step in carbohydrate metabolism, the reversible oxidative phosphorylation of glyceraldehyde-3-phosphate. Alternative splicing results in multiple transcript variants
CXCR5	It encodes a multi-pass membrane protein that belongs to the CXC chemokine receptor family. It is expressed in mature B-cells and Burkitt's lymphoma. Alternatively spliced transcript variants encode different isoforms
BLM	Its product is related to the RecQ subset of DExH box-containing DNA helicases and has both DNA-stimulated ATPase and ATP-dependent DNA helicase activities. Mutations causing Bloom syndrome delete or alter helicase motifs and may disable the 3′–5′ helicase activity
VAT1	Synaptic vesicles are responsible for regulating the storage and release of neurotransmitters in the nerve terminal. The encoded protein is an abundant integral membrane protein of cholinergic synaptic vesicles and is thought to be involved in vesicular transport
NTRK1	It encodes a member of the neurotrophic tyrosine kinase receptor (NTKR) family. Mutations in this gene have been associated with congenital insensitivity to pain, anhidrosis, self-mutilating behavior, mental retardation and cancer
$IWSS^3$	
GAPDH	It encodes a member of the glyceraldehyde-3-phosphate dehydrogenase protein family. Its product catalyzes an important energy-yielding step in carbohydrate metabolism, the reversible oxidative phosphorylation of glyceraldehyde-3-phosphate. Alternative splicing results in multiple transcript variants
TXK	It plays an overlapping role with ITK in iNKT cell development and function. ITK has a unique function in the survival of iNKT cells. ITK plus RLK inhibition may have therapeutic potential in Th1-mediated inflammatory diseases
GSTP1	Glutathione S-transferases are a family of enzymes for detoxification by catalyzing the conjugation of many hydrophobic and electrophilic compounds with reduced glutathione. Functionally different GSTP1 variant proteins that are thought to function in xenobiotic metabolism and play a role in susceptibility to cancer, and other diseases
WDDR1	Its encoded protein belongs to a subfamily of tyrosine kinase receptors with homology to Dictyostelium discoideum protein. Expression of this protein is restricted to epithelial cells, particularly in the lung, gastrointestinal tract, and brain. Alternatively spliced transcript variants encoding different isoforms have been described for this gene
IGFBP2	The protein encoded by this gene is one of six similar proteins that bind insulin-like growth factors I and II. The encoded protein can be secreted into the bloodstream or it can remain intracellular, interacting with many different ligands. Several transcript variants have been found for this gene
$IWSS^3$-MB	

feature; in evaluating the second feature, the classifier complexity is $O(g(2))$. In evaluating the last feature, IWSS-MB has a classifier complexity of $O(g(n\text{-}1))$, rather than $O(g(n))$. Thus, the overall time complexity is $O(g(1))+O(g(2))+\ldots+O(g(n-1))=\sum_{i=1}^{n-1} O(g(i))$, which is smaller than that of IWSS. d) More than one features are removed. According to the analysis of case c), we can obtain a lower time complexity.

Overall, from the above analysis, we can see that both SFS-MB and IWSS-MB can reduce the time complexity provided that there exists redundancy among the features, which is an inherent characteristic of microarray data [7,41]. We can also observe that a larger number of redundant features in the original feature space accompanies a greater reduction in time cost.

6. Conclusions

In this study, we proposed to integrate the Markov blanket technique into wrapper-based feature selection. Rather than evaluate all the candidate features using a classifier, the proposed method eliminates those features that are redundant to the selected features from the candidates. This helps reduce the number of candidate features and wrapper evaluations, and thus speed up the feature selection process. Extensive experimental comparisons between the cases without and with Markov blanket were conducted on ten microarray data with *k*-nearest-neighbor, naïve Bayes and C4.5. Experimental results show that Markov blanket-embedded methods greatly reduce the number of wrapper evaluations and the actual time costs without degrading the classification performance, and that the proposed method can generate a different feature subset to facilitate biomedical research. In addition, theoretical time and space complexity analysis also provide supports to our conclusions.

For the future research, we plan to work in the following four lines. First, Markov blanket criterion is one of the metrics that we can use to identity redundant features. So, our proposed method can be generalized to be a feature selection framework that can explicitly identify redundant features, and thus we can apply other filter methods rather than Markov blanket in our proposed methods. Second, although we tested the effectiveness and efficiency of the proposed methods merely on microarray data, they are general feature selectors that can be applied to other high throughput genomic data types (such as RNA-seq, ChIP-seq, and proteomics [43]) and even other fields (such as image classification and text categorization). Therefore, one of the future works involves testing the proposed approach in these alternate fields, and a slight modification of the relevance criteria of these algorithms may be required depending on the available samples. An alternative is to use a *t-test* to evaluate the goodness of a candidate feature [22]. Third, the proposed method is actually a hybrid feature selection strategy. We then plan to explore the integration of Markov blanket with other well-performing wrapper-based feature selectors, including the best incremental ranked subset (BIRS) [22], linear forward selection (LFS) [44], and best agglomerative ranked subset (BARS) [45], and test their effectiveness and efficiency. Fourth, our current approach eliminates these redundant features whose Markov blanket is the newly selected feature, and may ignore the interaction between the candidates feature with a subset of features. This motivates us to conduct further research to identify redundant features during the wrapper evaluations based on a subset of selected features rather than only according to the newly selected one.

A conflict of interest statement

None declared.

Acknowledgment

This work was supported partially by the China Postdoctoral Science Foundation (No. 2016M592046), the Fundamental Research Funds for the Central Universities (No. JZ2016HGBH1053), the “111 Project” of the Ministry of Education and State Administration of Foreign Experts Affairs (No. B14025), the Natural Science Foundation of China (No. 61472057, No. 61305064).

References

[1] P. Baldi, A. Long, A Bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes, Bioinformatics 17 (2001) 509–519.
[2] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.
[3] A. Wang, N. An, G. Chen, L. Li, G. Alterovitz, Improving PLS–RFE based gene selection for microarray data classification, Comput. Biol. Med. 62 (2015) 14–24.
[4] R. Somorjai, B. Dolenko, R. Baumgartner, Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions, Bioinformatics 19 (2003) 1484–1491.
[5] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (2007) 2507–2517.
[6] J. Hua, W. Tembe, E. Dougherty, Performance of feature-selection methods in the classification of high-dimension data, Pattern Recogn. 42 (2009) 409–424.
[7] W. Li, Y. Yang, How many genes are needed for a discriminant microarray data analysis, in: S.M. Lin, K.F. Johnson (Eds.), Methods of Microarray Data Analysis, Kluwer Academic Publishers, Springer, USA, 2002, pp. 137–149.
[8] A. Wang, N. An, G. Chen, L. Li, G. Alterovitz, Accelerating wrapper-based feature selection with K-Nearest-Neighbor, Knowl. -Based Syst. 83 (2015) 81–91.
[9] R. Luque-Baena, D. Urda, M. Claros, L. Franco, J. Jerez, Robust gene signatures from microarray data using genetic algorithms enriched with biological pathway keywords, J. Biomed. Inform. 49 (2014) 32–44.
[10] I. Gheyas, L. Smith, Feature subset selection in large dimensionality domains, Pattern Recogn. 43 (2010) 5–13.
[11] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn 46 (2002) 389–422.
[12] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. (2003) 1157–1182.
[13] A. Sharma, S. Imoto, S. Miyano, A top-r feature selection algorithm for microarray gene expression data, IEEE/ACM Trans, Comput. Biol. Bioinform 9 (2012) 754–764.
[14] M. Dash, H. Liu, Consistency-based search in feature selection, Artif. Intell. 151 (2003) 155–176.
[15] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, et al., Robust biomarker identification for cancer diagnosis with ensemble feature selection methods, Bioinformatics 26 (2010) 392–398.
[16] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Mach. Learn 53 (2003) 23–69.
[17] G. Brown, A. Pocock, M. Zhao, M. Luján, Conditional likelihood maximisation: a unifying framework for information theoretic feature selection, J. Mach. Learn. Res. 13 (2012) 27–66.
[18] R. Kohavi, G. John, Wrappers for feature subset selection, Artif. Intell. 97 (1997) 273–324.
[19] G. Chen, J. Chen, A novel wrapper method for feature selection and its applications, Neurocomputing 159 (2015) 219–226.
[20] S. Zheng, W. Liu, An experimental comparison of gene selection by Lasso and Dantzig selector for cancer classification, Comput. Biol. Med. 41 (2011) 1033–1040.
[21] Y. Sun, S. Todorovic, S. Goodison, Local-learning-based feature selection for high-dimensional data analysis, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1610–1626.
[22] R. Ruiz, J. Riquelme, J. Aguilar-Ruiz, Incremental wrapper-based gene selection from microarray data for cancer classification, Pattern Recogn. 39 (2006) 2383–2392.
[23] P. Bermejo, L. Ossa, J. Gámez, J. Puerta, Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking, Knowl. Based Syst. 25 (2012) 35–44.
[24] D. Koller, M. Sahami, Toward optimal feature selection, in: Proceedings of the 13th International Conference on Machine Learning, ACM, Bari, Italy, 1996, pp. 284–292
[25] A. Wang, N. An, J. Yang, L. Li. G. Chen, G. Alterovitz, Incremental wrapper based gene selection with Markov blanket, in: The ASE International Conference on Biomedical Computing, ASE, Cambridge, MA, USA, 2014, pp. 1–6
[26] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Trans. Neural Netw. 5 (1994) 537–550.
[27] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (2004) 1205–1224.
[28] M. Hall, L. Smith, Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper, in: Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference (FLAIRS), AAAI, Florida, USA, 1999, pp. 235–239
[29] P. Bermejo, J. Gámez, J. Puerta, On incremental wrapper-based attribute selection: experimental analysis of the relevance criteria, in: Conference on Information Processing and Mangement of Uncertainty (IPMU), Springer, Torremolinos, Spain,

2008, pp. 638–645
[30] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, A. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. 96 (1999) 6745–6750.
[31] J. Khan, J. Wei, M. Ringner, L. Saal, M. Ladanyi, F. Westermann, F. Berthold, et al., Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nat. Med. 7 (2001) 673–679.
[32] M. Shipp, K. Ross, P. Tamayo, A. Weng, J. Kutok, R. Aguiar, M. Gaasenbeek, et al., Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning, Nat. Med. 8 (2002) 68–74.
[33] D. Singh, P. Febbo, K. Ross, D. Jackson, J. Manola, C. Ladd, P. Tamayo, et al., Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2002) 203–209.
[34] L. Dyrskjot, T. Thykjaer, M. Kruhoffer, J. Jensen, et al., Identifying distinct classes of bladder carcinoma using microarrays, Nat. Genet. 33 (2003) 90–96.
[35] Y. Hippo, H. Taniguchi, S. Tsutsumi, N. Machidaet, et al., Global gene expression analysis of gastric cancer by oligonucleotide microarrays, Cancer Res. 62 (2002) 233–240.
[36] S. Piloto, T. Schilling, Ovo1 links Wnt signaling with N-cadherin localization during neural crest migration, Development 137 (2010) 1981–1990.
[37] T. MacDonald, K. Brown, B. LaFleur, K. Peterson, et al., Expression profiling of medulloblastoma: pdgfra and the RAS/MAPK pathway as therapeutic targets for metastatic disease, Nat. Genet. 29 (2001) 143–152.
[38] U. Braga-Neto, E. Dougherty, Is cross-validation valid for small-sample microarray classification?, Bioinformatics 20 (2004) 374–380.
[39] S. Singhi, H. Liu, Feature subset selection bias for classification learning, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, Pittsburgh, USA, 2006, pp. 849–856
[40] C. Ambroise, G. McLachlan, Selection bias in gene extraction on the basis of microarray gene-expression data, Proc. Natl. Acad. Sci. 99 (2002) 6562–6566.
[41] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, J. Bioinf. Comput. Biol. 3 (2005) 185–205.
[42] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[43] M. Grabherr, B. Haas, M. Yassour, J. Levin, D. Thompson, I. Amit, et al., Full-length transcriptome assembly from RNA-Seq data without a reference genome, Nat. Biotechnol. 29 (2011) 644–652.
[44] M. Gutlein, E. Frank, M. Hall, A. Karwath, Large-scale attribute selection using wrappers, in: IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, Nashville, USA, 2009, pp. 332–339
[45] R. Ruiz, J. Aguilar-Ruiz, J. Riquelme, Best agglomerative ranked subset for feature selection, in: JMLR Workshop and Conference Proceedings, New Challenges for Feature Selection (FSDM), JMIR, Belgium, 2008, pp. 148–162

Aiguo Wang was born in Anhui Province, China in 1986. He received his Ph.D. degree at Hefei University of Technology in 2015. He is now a post doctor with School of Computer and Information, Hefei University of Technology. His research interests include data mining, bioinformatics, and activity recognition.

Ning An was born in Gansu Province, China in 1971. He received his Ph.D. at Pennsylvania State University in 2002. He is now a professor with School of Computer and Information, Hefei University of Technology. His research interests include gerontechnology, healthcare informatics, and spatial information management.

Jing Yang was born in Anhui Province, China in 1979. She received the Ph.D. degree from Hefei University of Technology, in 2013. She is currently an associate professor with the School of Computer and Information, Hefei University of Technology. Her current research interests include artificial intelligence, data mining, and Bayesian network.

Guilin Chen was born in Anhui Province, China in 1965. He received M.Sc. in Anhui Normal University in 1985. He is now a professor at Chuzhou University. His research interests include Internet of Things, pervasive computing, and health informatics.

Lian Li was born in Shangdong Province, in 1951. He is a professor of Hefei University of Technology. His research interests include computational mathematics, grid computing, and social computing.

Gil Alterovitz is an assistant professor of Pediatrics at Harvard Medical School and is on the faculty of Boston Children's Hospital. He received his S.M. and Ph.D. in Electrical and Biomedical Engineering at Massachusetts Institute of Technology (MIT) in 2001 and 2006, respectively. His research interests include bioinformatics and proteomics.