STABILITY OF FEATURE SELECTION ALGORITHMS AND ENSEMBLE FEATURE SELECTION METHODS IN BIOINFORMATICS

PENGYI YANG, BING B. ZHOU, JEAN YEE-HWA YANG, ALBERT Y. ZOMAYA

SCHOOL OF INFORMATION TECHNOLOGIES AND SCHOOL OF MATHEMATICS AND STATISTICS, UNIVERSITY OF SYDNEY, NSW 2006, AUSTRALIA

{YANGPY,BBZ,ZOMAYA}@IT.USYD.EDU.AU; JEANY@MATHS.USYD.EDU.AU

1.1 INTRODUCTION

Feature selection is a key technique originated from the fields of artificial intelligence and machine learning [3, 10] in which the main motivation has been to improve sample classification accuracy [5]. Since the purpose is mainly on improving classification outcome, the design of feature selection algorithms seldom consider specifically on which features are selected. Due to the exponential growth of biological data in recent years, many feature selection algorithms have found to be readily applicable or with minor modification [32], for example, to identify potential disease associated genes from microarray studies [35], proteins from *mass spectrometry* (MS)-based proteomics studies [23], or *single nucleotide polymorphism* (SNP) from *genome wide association* (GWA) studies [37]. While sample classification accuracy is an important aspect in many of those biological studies such as discriminating cancer and normal tissues, the emphasis is also on the selected features as they represent interesting genes, proteins, or SNPs. Those biological features are often referred to as biomarkers and they often determine how the further validation studies should be designed and conducted.

One special issue arises from the application of feature selection algorithms in identifying potential disease associated biomarkers is that those algorithms may give unstable selection results [19]. That is, a minor perturbation on the data such as a different partition of data samples, removal of a few samples, or even reordering of the data samples may cause a feature selection algorithm to select a different set of features. For those algorithms with stochastic components, to simply rerun the algorithm with a different random seed may result in a different feature selection result.

The term *stability* and its counterpart *instability* are used to describe whether a feature selection algorithm is sensitive/insensitive to the small changes in the data and the settings of algorithmic parameters. The stability of a feature selection algorithm becomes an important property in many biological studies because biologists may be more confident on the feature selection results that do not change much on a minor perturbation on the data or a rerun of the algorithm. While this subject has been relatively neglected before, we saw a fast growing interests in recent years in finding different approaches for improving the stability of feature selection algorithms and different metrics for measuring them.

In this chapter, we provide a general introduction on stability of feature selection algorithms and review some popular ensemble strategies and evaluation metrics for improving and measuring feature selection stability. In Section 2, we categorize feature selection algorithms and illustrate some common causes of feature selection instability. In Section 3, we describe some popular methods for building ensemble feature selection algorithms and show the improvement of ensemble feature selection algorithms in terms of feature selection stability. Section 4 reviews some typical metrics that are used for evaluating the stability of a given feature selection algorithm. Section 5 concludes the chapter.

1.2 FEATURE SELECTION ALGORITHMS AND INSTABILITY

Feature selection stability has been a minor issue in many conventional machine learning tasks. However, the application of feature selection algorithms to bioinformatics problems, especially in disease associated biomarker identification, has arisen the specific interests in selection stability as evidenced by several recent publications [4, 17]. In this section, we first categorize feature selection algorithms according to the way they select features. Then we demonstrate the instability of different feature selection algorithms by three case studies.

1.2.1 Categorization of feature selection algorithm

From a computational perspective, feature selection algorithms can be broadly divided into three categories, namely *filter*, *wrapper*, and *embedded* according to their selection manners [10]. Figure 1.1 shows a schematic view of these categories.

Filter algorithms commonly rank/select features by evaluating certain types of association or correlation with class labels. They do not optimize the classification

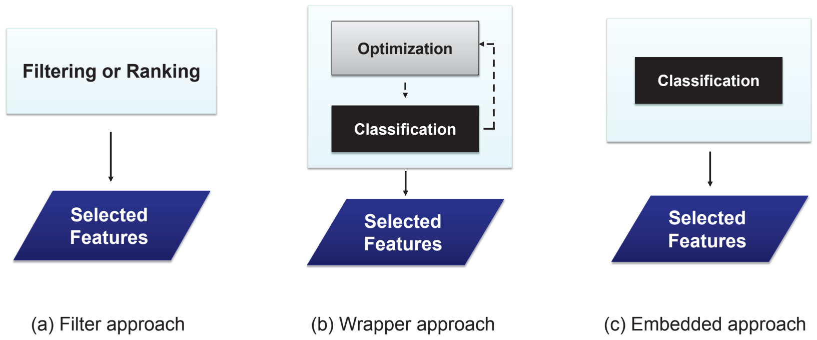


Figure 1.1 Categorization of feature selection algorithms. (a) Filter approach where feature selection is independent from the classification. (b) Wrapper approach where feature selection relies on an inductive algorithm for sample classification in an iterative manner. (c) Embedded approach where feature selection is performed implicitly by an inductive algorithm during sample classification.

accuracy of a given inductive algorithm directly. For this reason, filter algorithms are often computationally more efficient compared to wrapper algorithms. For numeric data analysis such as *differentially expressed* (DE) gene selection from microarray data or DE protein selection from mass spectrometry data, the most popular methods are probably t-test and its variants [33]. As for categorical data types such as disease associated SNP selection from GWA studies, the commonly used methods are χ^2-statistics, odds ratio, and increasingly the *ReliefF* algorithm and its variants [27].

Although filter algorithms often show good generalization on unseen data, they suffer from several problems. Firstly, filter algorithms commonly ignore the effects of the selected features on sample classification with respect to the specified inductive algorithm. Yet the performance of the inductive algorithm could be useful for accurate phenotype classification [21]. Secondly, many filter algorithms are univariate and greedy based. They assume that each feature contributes to the phenotype independently and thus evaluate each feature separately. A feature set are often determined by first ranking the features according to certain scores calculated by filter algorithms and then selecting the top-k candidates. However, the assumption of independence is invalid in biological systems and the selection results produced in this way are often suboptimal.

Compared to filter algorithms, wrapper algorithms have several advantages. Firstly, wrapper algorithms incorporate the performance of an inductive algorithm in feature evaluation, and therefore, likely to perform well in sample classification. Secondly, most wrapper algorithms are multivariate and treat multiple features as an unit for evaluation. This property preserves the biological interpretation of genes and proteins since they are linked by pathways and functioning in groups. A large number of wrapper algorithms have been applied to gene selection of microarray and protein selection of mass spectrometry. Those include evolution approaches such as *genetic algorithm* (GA) based selection [25, 24, 15], and greedy approaches such as incremental forward selection [30], and incremental backward elimination [28].

Despite their common advantages, wrapper approach often suffer from problems such as overfitting since the feature selection procedure is guided by an inductive algorithm that is fitted on a training data. Therefore, the features selected by wrapper approach may generalize poorly on new datasets if overfitting is not prevented. In addition, wrapper algorithms are often much slower compared to filter algorithms (by several orders of magnitude), due to their iterative training and evaluating procedures.

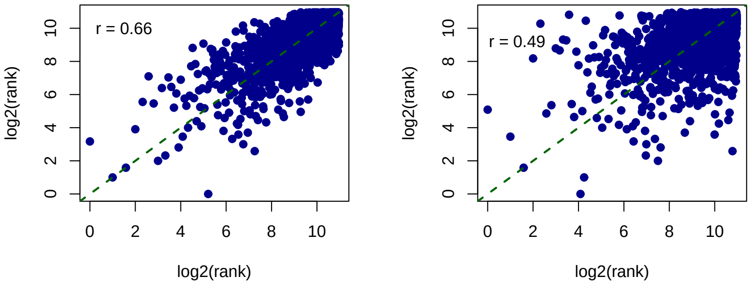
Embedded approach is somewhat between the filter approach and the wrapper approach where an inductive algorithm implicitly selects features during sample classification. Different from filter and wrapper approaches, the embedded approach relies on certain types of inductive algorithm and is therefore less generic. The most popular ones that applied for gene and protein selection are support vector machine based recursive feature elimination (SVM-RFE) [11] and random forest based feature evaluation [7].

1.2.2 Potential causes of feature selection instability

The instability of feature selection algorithms is typically amplified by small sample size which is common in bioinformatics applications. This is often demonstrated by applying bootstrap sampling on the original dataset and comparing the feature selection results from sampled datasets [1]. Beside the common cause of small sample size, the stability is also highly dependent on the types of feature selection algorithm in use. For example, wrapper based approaches rely on partitioning data into training and testing sets where training set is used to build the classification model and testing set is used for feature evaluation [9]. Therefore, a different partition of the training and testing sets may cause different feature selection results, and thus, instability. Feature selection algorithms using stochastic search such as GA based feature selection may give different selection results with a different random seed, initialization, and parameter setting. Some algorithms such as *ReliefF* based algorithms are sensitive to the sample order in feature selection from categorical dataset [36].

In this section, we demonstrate several common cases where the instability of feature selection is observed. We select typical filter, wrapper, and embedded feature selection algorithms for this demonstration. The case studies are classified according to the causes of feature selection instability.

1.2.2.1 Case study I: small sample size Small sample size is the common cause of feature selection instability. To demonstrate this effect, we applied bootstrap sampling on the colon cancer microarray dataset [2]. Colon cancer microarray dataset represents a typical microarray experiment where the normal samples and the tumor samples are compared. The dataset has 40 tumor samples and 22 normal ones obtained from colon tissue. Giving the number of genes measured (*i.e.* 2000) and the total number of samples (*i.e.* 62), it is a typical small sample size dataset with very high feature dimensionality.



(a) Moderated t-test

(b) SVM-RFE

Figure 1.2 Instability demonstration for a filter algorithm (moderated t-test) and an embedded algorithm (SVM-RFE) in feature selection on colon cancer microarray dataset [2]. (a) scatter plot of two runs of moderated t-test each calculated on a bootstrap sampling of the original dataset; (b) scatter plot of two runs of SVM-RFE each calculated on a bootstrap sampling of the original dataset. In each case, a Spearman correlation denoted as r is calculated.

Figure 1.2a shows the scatter plot of two runs of a filter algorithm known as moderated t-test [33]. Each run of moderated t-test is conducted on a bootstrap sampling from the original dataset with a different seeding. The x-axis and the y-axis are the ranking of genes (in logarithm of base 2) in the first and the second runs, respectively, plotted against each other. The most informative gene is ranked as 1, the second most informative one as 2, and so on. If the ranking of all genes remain the same in these two runs, they should form a diagonal line with a Spearman correlation of 1. However, it is clear that moderated t-test is highly unstable in ranking genes from small sample size dataset. A Spearman correlation (denoted as r) of 0.66 is observed from the two runs.

Figure 1.2b shows the result from using an embedded feature selection algorithm known as SVM-RFE. A SVM is built to evaluate features, and we eliminate 10% of total features in each iteration. The scatter plot of two runs of SVM-RFE each conducted on a separate bootstrap sampling indicates a low stability of a Spearman correlation of only 0.49. Therefore, similar to moderated t-test, SVM-RFE is also highly unstable in ranking genes from small sample size dataset.

1.2.2.2 Case study II: sample order dependency Feature selection results may be different even by changing the order of samples in the dataset. This may occur if the feature selection algorithm scores each feature by evaluating partial as opposed to all samples in the dataset, and the selection of the partial samples is dependent on the order of samples. This is best exemplified by using *ReliefF* based feature selection algorithms [29] for categorical feature selection.

Consider a GWA study consisting of N SNPs and M samples. Defining each SNP in the study as g_j and each sample as s_i where $j = 1 \ldots N$ and $i = 1 \ldots M$.

ReliefF algorithm ranks each SNP, by updating a weight function for each SNP at each iteration as follows:

$$W(g_j) = W(g_j) - D(g_j, s_i, h_k)/M + D(g_j, s_i, m_k)/M \tag{1.1}$$

where s_i is the ith sample from the dataset and h_k is the kth nearest neighbor of s with same the class label (called hit) while m_k is the kth nearest neighbor to s_i with a different class label (called $miss$). This weight updating process is repeated for M samples selected randomly or exhaustively. Therefore, dividing by M keeps the value of $W(g_j)$ to be in the interval [-1,1]. $D(.)$ is a difference function that calculates the difference between any two samples s_a and s_b for a given gene g:

$$D(g, s_a, s_b) = \begin{cases} 0 & \text{if } G(g, s_a) = G(g, s_b) \\ 1 & \text{otherwise} \end{cases} \tag{1.2}$$

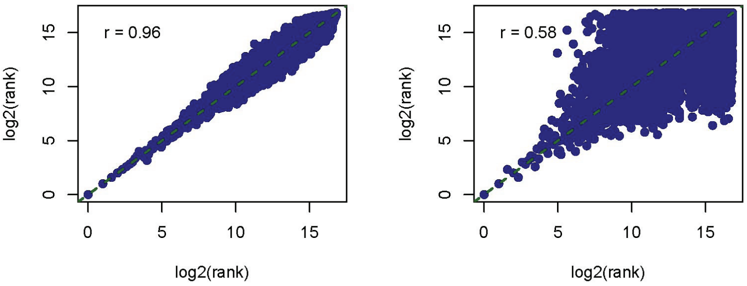
where $G(.)$ denotes the genotype of SNP g for sample s. The nearest neighbors to a sample are determined by the distance function, $MD(.)$, between the pairs of samples (denoted as s_a and s_b) which is also based on the difference function (Eq. 1.2):

$$MD(s_a, s_b) = \sum_{j=1}^{N} D(g_j, s_a, s_b) \tag{1.3}$$

Turned ReliefF (TuRF) proposed by Moore and White [27] aims to improve the performance of the *ReliefF* algorithm in SNP filtering by adding an iterative component. The signal-to-noise ratio is enhanced significantly by recursively removing the low-ranked SNPs in each iteration. Specifically, if the number of iteration of this algorithm is set to R, it removes the N/R lowest ranking (*i.e.*, least discriminative) SNPs in each iteration, where N is the total number of SNPs.

However, both *ReliefF* and TuRF are sensitive to the order of samples in the dataset due to the assignment of hit and $miss$ nearest neighbors of each sample. Since K nearest neighbors are calculated by comparing the distance between each sample in the dataset and the target sample s_i, a tie occurs when more than K samples have a distance equal or less than the Kth nearest neighbor of s_i. It is easy to show that a dependency on the sample order can be caused by using any tie breaking procedure which forces exactly K samples out of all possible candidates to be the nearest neighbors of s_i, which causes a different assignment of hit and $miss$ of nearest neighbors when the sample order is permuted.

We demonstrate the sample order dependency effect by using *ReliefF* and TuRF algorithms, respectively, on a GWA study of *age-related macular degeneration* (AMD) dataset [20]. In this experiment, we permuted the sample order in the original dataset and applied *ReliefF* and TuRF to the original dataset and the perturbed dataset for SNP ranking. The ranking of each SNP in the two runs are log-transferred and plotted against each other (Figure 1.3a,b). While such an inconsistency is relatively small for the *ReliefF* algorithm, it is enhanced through the iterative ranking procedure of TuRF. A Spearman correlation of only 0.58 is obtained from the original and the sample order perturbed dataset.



(a) *ReliefF*

(b) TuRF

Figure 1.3 Instability demonstration for *ReliefF* and TuRF algorithms. (a) scatter plot of two runs of *ReliefF* one on the original AMD dataset [20] whereas the other on a sample order perturbed dataset. (b) scatter plot of two runs of TuRF one on the original AMD dataset whereas the other on a sample order perturbed dataset. In each case, a Spearman correlation denoted as r is calculated.

1.2.2.3 Case study III: data partitioning

Typical wrapper algorithms generally build classification models and evaluate features using the models for data classification. For the purpose of building models and evaluating features, the dataset is partitioned into training and testing subsets where the training set is used with an inductive algorithm for creating classification models and the testing set is used for evaluating features using the models obtained from the training set. Note that the partition of dataset is often necessary since using the entire dataset for both model building and feature evaluation would overfit the model easily and produce ungeneralizable feature selection results. The feature selection result from wrapper algorithms could be unstable due to different splittings of data partition. Moreover, wrapper algorithms often rely on certain stochastic or heuristic algorithm (known as the search algorithm) to evaluate features in combination so as to reduce the large search space. Therefore, a different seeding or initialization of the search algorithm or a different parameter setting in heuristic search could also produce different feature selection results.

Here we demonstrate the instability in wrapper based feature selection algorithms using a wrapper of GA (genetic algorithm) with a k-nearest neighbor (kNN) as induction algorithm for feature selection (GA/kNN). Since the initial work by Li *et al.* [25], this configuration and its variants have become very popular in biomarker selection from high-dimensional data. We fix the neighbor size as $k = 3$ in all experiments, and the partition of dataset as 5-fold cross validation. The parameter setting of GA is also fixed to the default values as specified in Weka package [12]. Figure 1.4 shows two separate runs of GA/kNN wrapper algorithm each with a different 5-fold cross validation partitioning of colon microarray dataset [2] for model training and feature evaluation. After running GA/kNN, a gene is either selected or

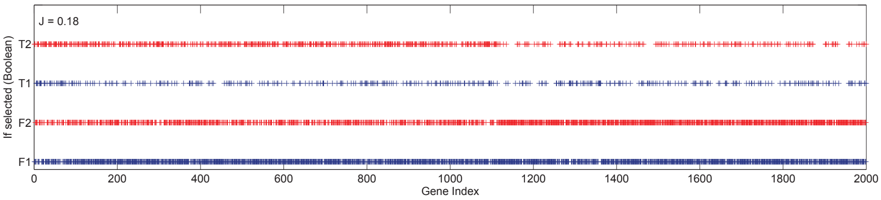


Figure 1.4 Instability demonstration for GA/kNN wrapper algorithm with colon cancer microarray dataset. The x-axis is the index of the 2000 genes in the colon cancer microarray dataset [2]. The y-axis is a boolean value indicating whether a gene is selected. Two separate runs of GA/kNN each with a different 5-fold cross validation partitioning of the dataset. For example, a cross is added to the x-axis of 200 and y-axis of T2 if the gene with index of 200 in the dataset is selected in the second run. A Jaccard set-based index denoted as *J* is calculated (see 1.4.2 for details).

unselected. If the algorithm is not sensitive to a different partitioning of the dataset, the genes selected in the first run should also be selected in the second run.

To quantify the concordance of the two runs in terms of the selected genes, we use a metric known as Jaccard set-based index (see 1.4.2 for details) to compute the similarity of the two runs. A Jaccard set-based index of 0.18 indicates a low reproducibility of the GA/kNN wrapper algorithm on feature selection. Therefore, the algorithm is highly unstable when the dataset is partitioned differently.

1.2.3 Remark on feature selection instability

Although we have demonstrated some common causes of feature selection instability separately, they should not be considered independently. For example, a wrapper algorithm could suffer from a combination effect of small sample size and partition of the dataset. A *ReliefF* based algorithm could suffer from the sample order perturbation and a different size of k used to determine nearest neighbors. The SVM-RFE algorithm could suffer from small sample size and a different step size of recursive feature elimination.

There are several possible ways to improve stability of feature selection algorithms such as using prior information and knowledge, feature grouping, and ensemble feature selection. We will focus specifically on ensemble feature selection which is introduced in Section 1.3. A review for several other approaches can be found in [13].

The stability of feature selection algorithms is generally assessed by using certain metric. We have used Spearman correlation and Jaccard set-based index in our case studies. The details of those metrics and many others are described in Section 1.4.

1.3 ENSEMBLE FEATURE SELECTION ALGORITHMS

The purpose of composing ensemble feature selection algorithms is manyfold. Generally, the goals are to improve feature selection stability or sample classification accuracy or both at the same time as demonstrated in numerous studies [1, 26, 16]. In many cases, other aspects such as to identify important features or to extract feature interaction relationships could also be achieved in a higher accuracy using ensemble feature selection algorithms as compared to their single versions.

Depending on the type of feature selection algorithm, there may be many different ways to compose an ensemble feature selection algorithm. Here we describe two most commonly used approaches.

1.3.1 Ensemble based on data perturbation

The first approach is based on data perturbation. This approach has been extensively studies and utilized as can be viewed in the literature [4, 1, 36]. The idea is built on the successful experience in ensemble classification [8] and it has been found to stabilize the feature selection result. For example, a bootstrap sampling procedure can be used for creating an ensemble of filter algorithms each gives a slightly different ranking of genes. The consensus is then obtained through combining those ranking lists. It is natural to understand that beside bootstrap sampling many other data perturbation methods (such as random spacing etc.) can also be used to create multiple versions of the original dataset in the same framework. A schematic illustration of this class of methods is shown in Figure 1.5.

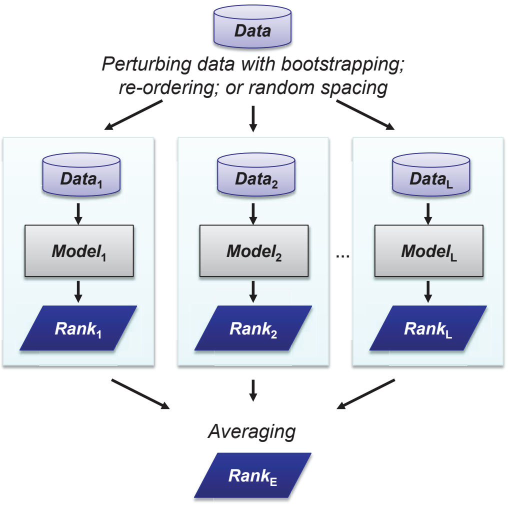


Figure 1.5 Schematic illustration of an ensemble of filters using data perturbation approach.

1.3.2 Ensemble based on different data partitioning

The second approach is based on partition the training and testing data differently which is specifically for wrapper based feature selection algorithms. That is, data that are used for building the classification model and the data that are used for feature evaluation are partitioned using multiple cross validations (or any other random partitioning procedures). The final feature subset is determined by calculating the frequency of each feature been selected from each partitioning. If a feature is selected more than a given threshold, it is then included into the final feature set.

A schematic illustration of this method is shown in Figure 1.6. This methods is firstly described in [9] where a *forward feature selection* (FFS) wrapper and a *backward feature elimination* (BFE) wrapper are shown to benefit from this ensemble approach.

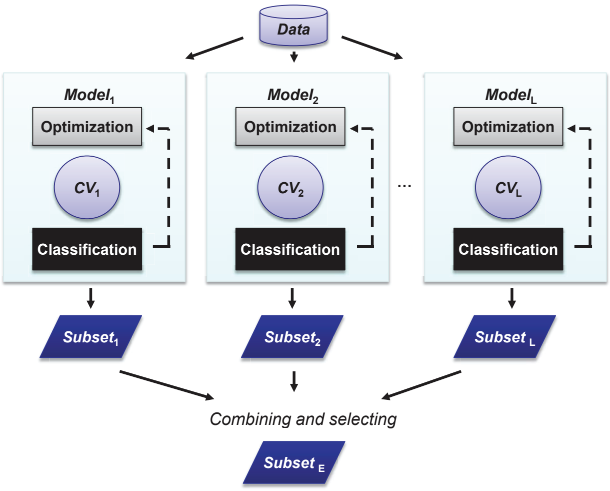


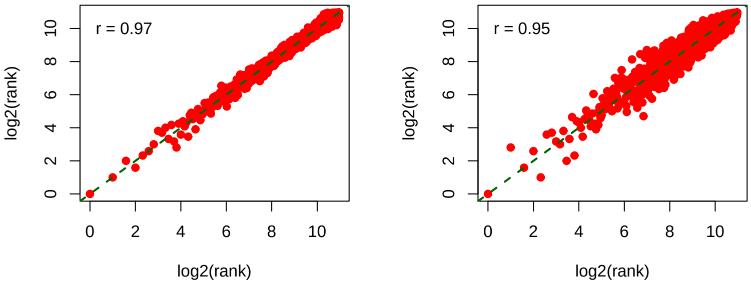
Figure 1.6 Schematic illustration of an ensemble of wrappers using different partitions of an internal cross validation for feature evaluation.

Beside using a different data partitioning, for stochastic optimization algorithms such as GA or *particle swarm optimization* (PSO), ensemble could also be achieved by using different initializations or different parameter settings. For wrappers such as FFS or BFE, a different starting point in the feature space could result in a different selection result. Generally, bootstrap sampling or other random spacing approaches can also be applied to wrapper algorithms for creating ensembles.

1.3.3 Performance on feature selection stability

We continue the examples in Section 1.2.2 and evaluate the performance of ensemble feature selection algorithms in terms of feature selection stability.

1.3.3.1 For small sample size problem For the small sample size problem, we evaluated the ensemble version of moderated t-test and the ensemble version of SVM-RFE (Figure 1.7a,b). Each ensemble run of moderated t-test was generated by aggregating (using averaging) 50 individual runs of bootstrap sampling from the original colon cancer dataset [2]. An ensemble of 50 individual runs were combined and plotted against another ensemble of 50 individual runs, with each individual run conducted with a different bootstrap seeding. The same procedure was also used to create the ensemble of SVM-RFE.



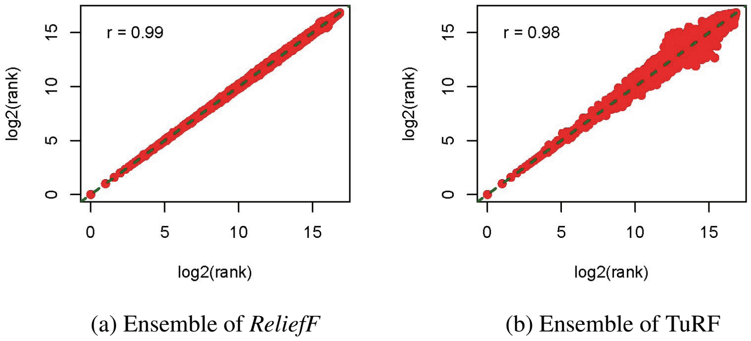
(a) Ensemble of moderated t-test (b) Ensemble of SVM-RFE

Figure 1.7 Ensemble feature selection algorithms for small sample size. (a) scatter plot of two runs of ensemble of moderated t-test each calculated on and combined from 50 bootstrap sampling of the original colon cancer dataset [2]; (b) scatter plot of two runs of ensemble of SVM-RFE each calculated on and combined from 50 bootstrap sampling of the original colon cancer dataset. Multiple ranking lists are combined by averaging. In each case, a Spearman correlation denoted as r is calculated.

It appears that the ensemble of moderated t-test is much better in terms of feature selection stability, with most of gene rankings clustering close to the diagonal line. A Spearman correlation of 0.97 is obtained compared to 0.66 from the single runs (Figure 1.2a). Similarly, the ensemble of SVM-RFE is able to increase the Spearman correlation from 0.49 to 0.95.

1.3.3.2 For sample order dependency problem The ensemble of *ReliefF* and TuRF were created by using random sample re-ordering for generating multiple SNP ranking lists and the consensus is obtained by simple averaging. Figure 1.8 shows the ensemble version of *ReliefF* and TuRF algorithms where an ensemble size of 50 is used.

It is clear that the ensemble approach for both *ReliefF* and TuRF algorithms can improve their consistency on feature selection when the sample order is perturbed. The improvement is especially encouraging for TuRF since two runs of a single TuRF only give a Spearman correlation of 0.58 (Figure 1.3b) whereas the ensembles of TuRF improve the Spearman correlation to 0.98 (Figure 1.8b).



(a) Ensemble of *ReliefF*

(b) Ensemble of TuRF

Figure 1.8 Ensemble feature selection algorithms for sample order dependency. (a) scatter plot of two runs of ensemble of *ReliefF* each calculated and combined from 50 sample order perturbed datasets from the original AMD dataset [20]; (b) scatter plot of two runs of ensemble of TuRF each calculated and combined from 50 sample order perturbed datasets from the original AMD dataset. In each case, a Spearman correlation denoted as r is calculated.

1.3.3.3 For data partitioning problem We conducted two separate runs of an ensemble of GA/kNN wrapper algorithm (an ensemble size of 50 is used) each with a different 5-fold cross validation partitioning of the colon cancer dataset [2]. Figure 1.9 shows the concordance of two ensemble of GA/kNN. The Jaccard set-based index increases from 0.18 (Figure 1.4) to 0.59, indicating that the ensemble version of GA/kNN can generate much more consistent feature selection results compared to the original GA/kNN algorithm.

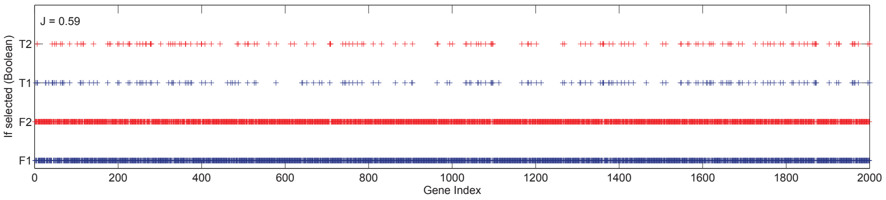
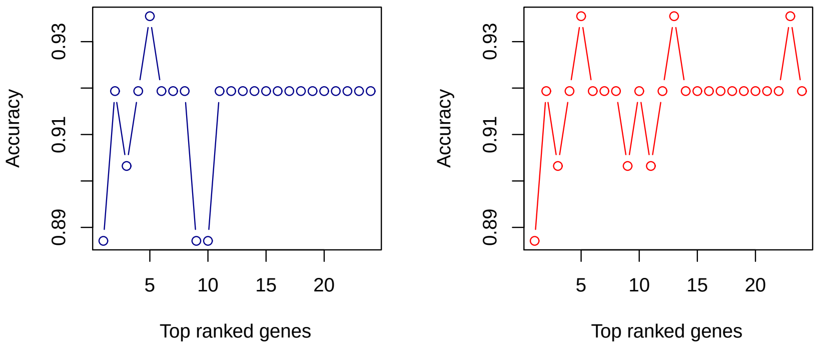


Figure 1.9 Ensemble feature selection algorithms for data partitioning. The x-axis is the index of the 2000 genes in the colon cancer microarray dataset [2]. The y-axis is a boolean value indicating whether a gene is selected. Two separate runs of ensemble of GA/kNN (an ensemble of 50) each with a different 5-fold cross validation partitioning of the dataset. For example, a cross is added to the x-axis of 200 and y-axis of T2 if the gene with index of 200 in the dataset is selected in the second run of the ensemble of GA/kNN. A Jaccard set-based index denoted as J is calculated (see 1.4.2 for details).

1.3.4 Performance on sample classification

Besides improving stability, another goal is to achieve higher classification accuracy by using ensemble feature selection approach [1]. Here we tested the classification accuracy using the genes selected by moderated t-test from colon cancer microarray dataset [2], and compare those results with its ensemble version. The classification accuracy was calculated using a 10-fold cross validation with a k-nearest neighbor classifier ($k = 3$).



(a) Moderated t-test

(b) Ensemble of moderated t-test

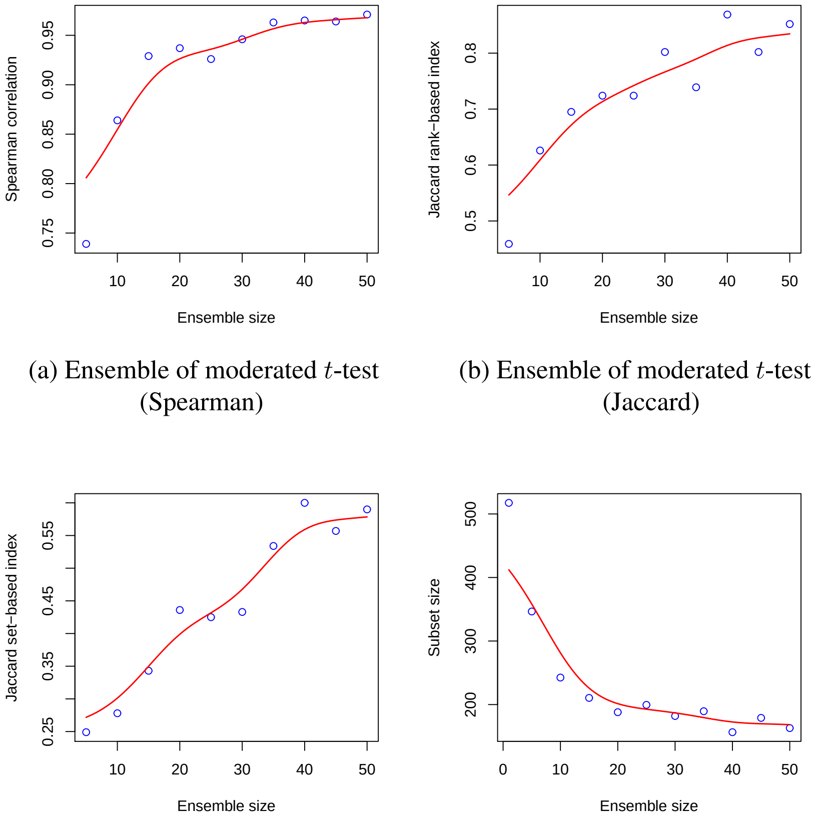
Figure 1.10 Sample classification accuracy using genes selected from colon cancer microarray dataset [2] by (a) moderated t-test, and (b) ensemble of moderated t-test.

From Figure 1.10, we observe that genes selected using the ensemble approach produce a minor improvement on sample classification as compared to the single approach. Since the sample size of the dataset is small, we anticipate that a greater improvement on sample classification may be achieved by using a dataset with larger sample size.

1.3.5 Ensemble size

For ensemble feature selection, the choice of ensemble size may affect the performance on feature selection and stability. In this subsection, we evaluate the effect of different ensemble sizes on feature selection stability. All the evaluation are done on colon cancer microarray dataset [2]. Several different evaluation metrics are used to assess the stability. Those metrics are described in details in Section 1.4.

From Figure 1.11a,b, we can see that larger ensemble size of the moderate t-test corresponds to higher feature selection stability in terms of both Spearman correlation and Jaccard rank-based index. Similar effect is also observed for the ensemble of



(a) Ensemble of moderated t-test (Spearman)

(b) Ensemble of moderated t-test (Jaccard)

(c) Ensemble of GA/kNN (Jaccard)

(d) Ensemble of GA/kNN (Subset size)

Figure 1.11 Ensemble size and its effects. (a) ensemble size of moderated t-test and its effect on feature selection stability measured by Spearman correlation; (b) ensemble size of moderated t-test and its effect on feature selection stability measured by Jaccard rank-based index; (c) Ensemble size of GA/kNN and its effect on feature selection stability measured by Jaccard set-based index; (d) Ensemble size of GA/kNN and its effect on selected feature subset size.

GA/kNN where increasing ensemble size results in higher feature selection stability as indicated by Jaccard set-based index (Figure 1.11c).

The size of the selected gene subsets has been used by many studies as an evaluation standard for wrapper algorithms [30]. Specifically, without sacrificing the performance, e.g. classification accuracy, small gene subsets are preferred. We observe that the larger the ensemble size of GA/kNN the smaller the identified gene subset as shown in Figure 1.11d.

1.3.6 Some key aspects in ensemble feature selection algorithms

There are several key aspects that may be of interests in designing ensemble feature selection algorithms. Firstly, how to create multiple models is important and determines the quality of the final feature selection results. An ensemble of certain feature selection algorithm may be created by using bootstrap sampling, random data partitioning, parameter randomization, or the combination of several. However, some ensemble approaches are specific to certain types of feature selection algorithms. For example, we can use sample order perturbation for creating an ensemble of *ReliefF* algorithm, but this approach will not help on a t-test filter. Similarly, we can not use the data partitioning approach for the filter-based feature selection algorithms as the classification is independent from the feature selection procedure.

The ensemble approach attempts to improve feature selection result by increasing model complexity. Why the added complexity may improve the feature selection result leads to the second key aspect known as diversity which is intensively studied in designing ensemble of classification algorithms [34]. However, to our knowledge this aspect has not been systematically studied in ensemble feature selection. Therefore, it is interesting to evaluate relationship between the performance on sample classification, the feature selection stability, and the diversity of ensemble models in ensemble feature selection algorithms.

The third key aspect on ensemble feature selection algorithm is on designing appropriate methods for combining multiple ranking lists or feature subsets. Some initial work has been done in this aspect [6] but the main approach still remains to be simple averaging. More sophisticated approach is clearly welcomed for improving the finial feature selection result.

Several other aspects such as model selection and model averaging that has been studied in ensemble classification could also be applied to study ensemble feature selection algorithms.

1.4 METRICS FOR STABILITY ASSESSMENT

Stability metrics are used to assess the stability of multiple feature selection results. A feature selection algorithm is often considered as stable in terms of feature selection if the selected features are consistent from multiple runs of the algorithm with variants of the original dataset. Depending on the types of the feature selection algorithm, multiple variants of the original dataset can be obtained by perturbing the original dataset in certain way such as bootstrapping, random partitioning, or re-ordering etc. We denote each feature selection results as $\mathcal{F}_i, \ (i = 1...L)$ where L is the number of times the selection are repeated. To assess the stability, it is common to perform a pairwise comparison of each selection result with others and average the assessment with respect to the number of comparisons [9]. Formally, this procedure can be expressed as follows:

$$\overline{S}_p = \frac{2}{L(L-1)} \sum_{i=1}^{L} \sum_{j=i+1}^{L} S(\mathcal{F}_i, \mathcal{F}_j) \tag{1.4}$$

where $\overline{S}_p$ is the assessment score of stability from averaged pairwise comparisons. $\mathcal{F}_i$ and $\mathcal{F}_j$ are the ith feature selection result and the jth feature selection result generated from different runs of a feature selection algorithm. $S(.)$ is a stability assessment metric which could be defined differently according to the type of feature selection algorithm and ones interests or emphasis in assessment.

1.4.1 Rank-based stability metrics

Rank-based metrics are used to assess the stability of multiple ranking lists in which the features are ranked based on certain evaluation criteria of a feature selection algorithm. Filter algorithms that produce a “goodness" score for each feature can be assessed using rank-based metrics whereas wrapper algorithms that generate a subset of features instead of ranking the features may not be assessed properly using rank-based metrics but require set-based stability metrics which will be introduced in Section 1.4.2.

Within the rank-based metrics, there are mainly two sub-categories depending on whether the full ranking list or a partial ranking list is considered. For the full ranking list, one assess the stability based on the rank of all features whereas for the partial list, a threshold is specified and only those that pass the threshold are used for stability assessment.

1.4.1.1 Full ranking metrics The most widely used metric for full ranking list is probably *Spearman correlation* [19, 18, 31]. For stability assessment, it is applied as follows:

$$S_S(\mathcal{R}_i, \mathcal{R}_j) = 1 - \sum_{\tau=1}^{N} \frac{6(r_i^\tau - r_j^\tau)^2}{N(N^2-1)} \tag{1.5}$$

where $S_S(\mathcal{R}_i, \mathcal{R}_j)$ denotes computing stability score on ranking lists $\mathcal{R}_i$ and $\mathcal{R}_j$ using Spearman correlation. N is the total number of features, and τ is an index goes through the first feature to the last one in the dataset. r_i^τ denotes the rank of the τth feature in the ith ranking list.

Spearman correlation ranges between -1 to 1 with 0 indicates no correlation and 1 or -1 indicate a perfect positive or negative correlation, respectively. For feature selection stability assessment, the higher (in positive value) the Spearman correlation the more consistent the two ranking lists, and therefore, the more stable the feature selection algorithm.

1.4.1.2 Partial ranking metrics In contrast to the full ranking metrics, partial ranking metrics require to pre-specify a threshold and consider only features that pass the threshold [14]. For example, the *Jaccard rank-based index* is a typical partial ranking metric used for assessing stability of feature selection algorithm in several studies [19, 31]. Here, one need to make a decision on using what percentage of top ranked features or simply how many top ranked features for stability assessment. Let us use top k features in each list. The Jaccard rank-based index can be computed as follows:

$$S_J^k(\mathcal{R}_i, \mathcal{R}_j) = \sum_{\tau=1}^{N} \frac{I(r_i^\tau \leqslant k \wedge r_j^\tau \leqslant k)}{2k - I(r_i^\tau \leqslant k \wedge r_j^\tau \leqslant k)} \tag{1.6}$$

where $S_J^k(\mathcal{R}_i, \mathcal{R}_j)$ denotes computing stability score on ranking lists $\mathcal{R}_i$ and $\mathcal{R}_j$ using Jaccard rank-based index with top k ranked features. $I(.)$ is an indicator function which gives 1 if an evaluation is true or 0 otherwise. As defined before, r_i^τ denotes the rank of the τth feature in the ith ranking list. $\wedge$ is the logic `and`.

What the above function essentially does is to find the intersection and the union of the top k features from ranking list i and ranking list j, and then compute the ratio of the intersection over union. Clearly, if the top k features in both ranking lists are exactly the same, the intersection and the union of them will be the same and therefore the ratio is 1 (perfect stability). Otherwise the ratio will be smaller than 1 and reaches 0 when none of the top k features in the two ranking lists is the same (no stability). Note that Jaccard rank-based metric is undefined when $k = 0$ and it is always 1 if all features are considered ($k = N$). Both cases are meaningless in the context of feature selection. In other words, we need to specify a meaningful threshold k that fulfill the inequality $0 < k < N$.

Another partial ranking metric is the *Kuncheva index* proposed by Kunckeva [22]:

$$S_I^k(\mathcal{R}_i, \mathcal{R}_j) = \sum_{\tau=1}^{N} I(r_i^\tau \leqslant k \wedge r_j^\tau \leqslant k) \frac{1 - k^2/N}{k - k^2/N} \tag{1.7}$$

where N is the total number of features, $I(r_i^\tau \leqslant k \wedge r_j^\tau \leqslant k)$ as defined before computes the number of features in common in the top k features of the ranking lists of i and j, and N is the total number of features.

Similar to the Jaccard rank-based index, the Kuncheva index looks at the intersection of the top k features in the two ranking lists i and j. However, instead of normalizing the intersection using the union of the two partial lists as in the Jaccard rank-based index, the Kuncheva index normalizes the intersection using the length of the list (that is, k) and correct for the chance of selecting common features at random among two partial lists with the term k^2/N. This is done by incorporating the total number of features N in the ranking list to the metric which takes into account the ratio of the number of features considered (k) and the total number of feature (N) in computing the index. The Kuncheva index is in the range of -1 to 1 with a greater value suggesting a more consistent feature selection results in the two runs. Similar to he Jaccard rank-based index, the Kuncheva index is undefined at both $k = 0$ and $k = N$, which in meaningless in practice and is often ignored.

1.4.2 Set-based stability metrics

For algorithms that directly selecting features instead of ranking features, a boolean value is produced indicating whether a feature is included or excluded in the feature selection result. In such a scenario, a set-based metric is more appropriate to evaluate stability of the feature selection result.

The most common metric in this category is the *Hamming index* which is adopted by Dunne *et al.* [9] for evaluating the stability of a few wrapper algorithms in feature selection. Assuming the feature selection results of two independently runs of a feature selection algorithm produces two boolean lists $\mathcal{M}_i$ and $\mathcal{M}_j$ in which an “1" indicates that a feature is selected and a “0" indicates that a feature is excluded. The stability of the algorithm can be quantified as follows:

$$S_H(\mathcal{M}_i, \mathcal{M}_j) = 1 - \sum_{\tau=1}^{N} \frac{|m_i^\tau - m_j^\tau|}{N} \tag{1.8}$$

where m_i^τ and m_j^τ denote the value of the τth position in the boolean list of i and boolean list of j, respectively. Those values could either be 0 or 1. N as before is the total number of features in the dataset.

If same features are included or excluded in the two boolean lists, the term $\sum_{\tau=1}^{N} \frac{|m_i^\tau - m_j^\tau|}{N}$ will be 0 which will give a Hamming index of 1. On the contrary, if the feature selection results are exactly opposite to each other, the term $\sum_{\tau=1}^{N} \frac{|m_i^\tau - m_j^\tau|}{N}$ will be 1 and the Hamming index will be 0.

Besides the Hamming index, the Jaccard index could also be applied for evaluating the stability of set-based feature selection results. We refer to it as the *Jaccard set-based index* so as to differentiate it from the Jaccard rank-based index. The Jaccard set-based index is defined as follows:

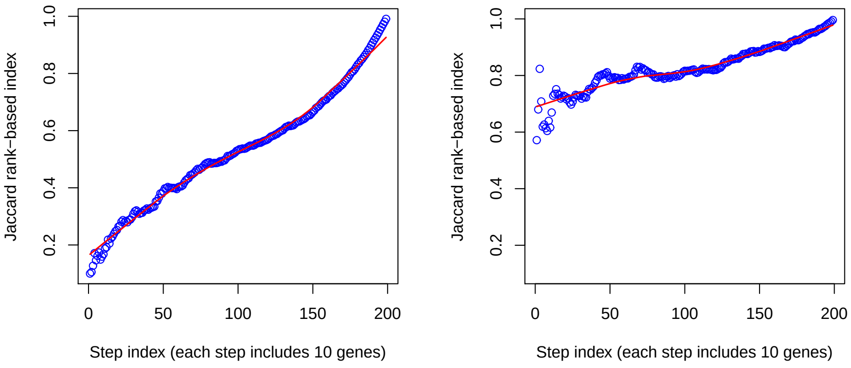
$$S_J(\mathcal{M}_i, \mathcal{M}_j) = \sum_{\tau=1}^{N} \frac{m_i^\tau \wedge m_j^\tau}{m_i^\tau \vee m_j^\tau} \tag{1.9}$$

where m_i^τ and m_j^τ as before denote the value of the τth position in the boolean list of i and boolean list of j. The term $m_i^\tau \wedge m_j^\tau$ over the sum of total number of features N gives the intersection of selected features in the two boolean list, whereas the term $m_i^\tau \vee m_j^\tau$ over the sum of total number of features gives the union of selected features.

1.4.3 Threshold in stability metrics

Depending on the feature selection algorithm and the biological questions, it may be more interesting to look at only the top ranked genes from a ranking list instead of considering all genes. This is generally true in cancer studies where only a subset of top ranked genes will be selected for follow up validation. In such a case, metrics that rely on a predefined threshold for calculation are often applied to study the stability of the feature selection results. The question here is on what threshold to use (say top 100 genes or top 500). Realize that a different threshold may lead to a different conclusion on stability.

Figure 1.12 shows the stability evaluation across multiple thresholds of Jaccard rank-based index. In particular, Figure 1.12a is the result using SVM-RFE and Figure 1.12b is the result using ensemble of SVM-RFE all with colon cancer microarray dataset [2]. Genes are ranked by the score from SVM-RFE or ensemble of SVM-RFE, respectively. We applied the thresholds of top 10, 20, ..., 2000 genes with a step of 10 genes for calculating the Jaccard rank-based index using bootstrap sampling datasets.



(a) SVM-RFE (b) Ensemble of SVM-RFE

Figure 1.12 Evaluation of using different thresholds for computing stability. SVM-RFE and its ensemble are run on bootstraps on colon cancer microarray dataset [2], respectively, and the stability value in terms of Jaccard rank-based index are calculated using thresholds from top 10 genes to top 2000 genes (that is, all genes) with a step of 10 (thus, 200 steps).

It is clear that the ensemble of SVM-RFE demonstrates much higher stability especially at the very top of the ranking lists. The stability according to Jaccard rank-based index is around 0.7 for the ensemble approach whereas for the single version of SVM-RFE it is less than 0.2. One important observation is that as more genes are included for calculation, the difference of the Jaccard rank-based index between the ensemble and the single approaches become smaller and eventually become 0 when all genes are included for calculation. Therefore, it may be most informative to compare the very top of the ranking lists when using the Jaccard rank-based index, whereas the comparison of a long list using the Jaccard rank-based index could be meaningless as both of them will have a value close to 1.

1.4.4 Remark on metrics for stability evaluation

It is generally unnecessary or even impossible to determine which metric is the best one for evaluating stability across all scenarios [19, 14]. In practise, depending on the type of feature selection algorithm, certain metric may appear to be more appropriate.

Sometimes, different metrics could be applied to the same selection results and they may help to determine different properties of a feature selection algorithm in terms of stability.

Since different metrics may score a feature selection algorithm differently, demonstrating that an algorithm performs more stable than other algorithms across multiple metrics is desirable for designing method to improve stability of feature selection.

1.5 CONCLUSIONS

Stability of feature selection algorithms has become an important research topic in bioinformatics where the selected features have important biological interpretations. Ensemble feature selection approach has been a general and promising solution in many scenarios where the performance of a single feature selection algorithm is highly unstable. In this chapter, we categorized feature selection algorithms into three types and demonstrated their instability in different scenarios. We focused on the ensemble feature selection algorithms and compared their performance with their corresponding single versions. Several metrics that are commonly used for assessing feature selection stability are introduced and used in our comparison.

Ensemble feature selection algorithms appear to be much more stable in terms of generating feature subset or feature ranking list. However, factors such as the size of the ensemble, the metric used for assessment, and the threshold used by some metrics should be taken into consideration when designing and comparing ensemble feature selection algorithms. We believe that ensemble feature selection algorithms are useful tools in bioinformatics applications where the goal is both on accurate sample classification and biomarker identification.

Acknowledgement

Pengyi Yang is supported by the NICTA International Postgraduate Award (NIPA) and the NICTA Research Project Award (NRPA).

References

1. T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3):392–398, 2010.
2. U. Alon, N. Barkai, DA Notterman, K. Gish, S. Ybarra, D. Mack, and AJ Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, pages 6745–6750, 1999.
3. A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.
4. A.L. Boulesteix and M. Slawski. Stability and aggregation of ranked gene lists. *Briefings in Bioinformatics*, 10(5):556–568, 2009.
5. M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(3):131–156, 1997.
6. RP DeConde, S. Hawley, S. Falcon, N. Clegg, B. Knudsen, and R. Etzioni. Combining results of microarray experiments: a rank aggregation approach. *Statistical Applications in Genetics and Molecular Biology*, 5:Article15, 2006.
7. R. Díaz-Uriarte and S.A. De Andres. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1):3, 2006.
8. T.G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 1–15. Springer-Verlag, 2000.

9. K. Dunne, P. Cunningham, and F. Azuaje. Solutions to instability problems with sequential wrapper-based approaches to feature selection. *Technical Report TCD-CS-2002-28, Department of Computer Science, Trinity College, Dublin, Ireland*, 2002.
10. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.
11. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, 2002.
12. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
13. Z. He and W. Yu. Stable feature selection for biomarker discovery. *Computational Biology and Chemistry*, 34(4):215–225, 2010.
14. I.B. Jeffery, D.G. Higgins, and A.C. Culhane. Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data. *BMC Bioinformatics*, 7(1):359, 2006.
15. T. Jirapech-Umpai and S. Aitken. Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC Bioinformatics*, 6(1):148, 2005.
16. K. Jong, J. Mary, A. Cornuéjols, E. Marchiori, and M. Sebag. Ensemble feature ranking. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 267–278. Springer-Verlag New York, Inc., 2004.
17. G. Jurman, S. Merler, A. Barla, S. Paoli, A. Galea, and C. Furlanello. Algebraic stability indicators for ranked lists in molecular profiling. *Bioinformatics*, 24(2):258, 2008.
18. A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 218–225. IEEE, 2005.
19. A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems*, 12(1):95–116, 2007.
20. R.J. Klein, C. Zeiss, E.Y. Chew, J.Y. Tsai, R.S. Sackler, C. Haynes, A.K. Henning, J.P. SanGiovanni, S.M. Mane, S.T. Mayne, et al. Complement factor h polymorphism in age-related macular degeneration. *Science*, 308(5720):385–389, 2005.
21. R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.
22. L.I. Kuncheva. A stability index for feature selection. In *Proceedings of the 25th IASTED International Multi-Conference: Artificial Intelligence and Applications*, pages 390–395. ACTA Press, 2007.
23. I. Levner. Feature selection and nearest centroid classification for protein mass spectrometry. *BMC Bioinformatics*, 6(1):68, 2005.
24. L. Li, D.M. Umbach, P. Terry, and J.A. Taylor. Application of the ga/knn method to seldi proteomics data. *Bioinformatics*, 20(10):1638, 2004.
25. 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):1131–1142, 2001.

26. B. Liu, Q. Cui, T. Jiang, and S. Ma. A combinational feature selection and ensemble neural network method for classification of gene expression data. *BMC Bioinformatics*, 5(1):136, 2004.

27. J. Moore and B. White. Tuning relieff for genome-wide genetic analysis. *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pages 166–175, 2007.

28. G. Potamias, L. Koumakis, and V. Moustakis. Gene selection via discretized gene-expression profiles and greedy feature-elimination. *Methods and Applications of Artificial Intelligence*, pages 256–266, 2004.

29. M. Robnik-Šikonja and I. Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, 53(1):23–69, 2003.

30. 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):2383–2392, 2006.

31. Y. Saeys, T. Abeel, and Y. Peer. Robust feature selection using ensemble feature selection techniques. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases-Part II*, pages 313–325. Springer-Verlag, 2008.

32. Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

33. GK Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3:Article3, 2004.

34. A. Tsymbal, M. Pechenizkiy, and P. Cunningham. Diversity in search strategies for ensemble feature selection. *Information Fusion*, 6(1):83–98, 2005.

35. E.P. Xing, M.I. Jordan, and R.M. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 601–608. Morgan Kaufmann Publishers Inc., 2001.

36. P. Yang, J. Ho, Y. Yang, and B. Zhou. Gene-gene interaction filtering with ensemble of filters. *BMC Bioinformatics*, 12(Suppl 1):S10, 2011.

37. K. Zhang, Z.S. Qin, J.S. Liu, T. Chen, M.S. Waterman, and F. Sun. Haplotype block partitioning and tag snp selection using genotype data and their applications to association studies. *Genome Research*, 14(5):908, 2004.