A review of ensemble methods in bioinformatics: *
Including stability of feature selection and ensemble feature selection methods

(updated on 28 Sep. 2016)

Pengyi Yang[1,2,3,*], *Yee Hwa Yang*[2], *Bing B. Zhou*[1,4] *and Albert Y. Zomaya*[1,4]

[1] School of Information Technologies, University of Sydney, NSW 2006, Australia
[2] School of Mathematics and Statistics, University of Sydney, NSW 2006 Australia
[3] NICTA, Australian Technology Park, Eveleigh, NSW 2015, Australia
[4] Centre for Distributed and High Performance Computing, University of Sydney, NSW 2006, Australia

* Corresponding author. Pengyi Yang, School of Information Technologies (J12), University of Sydney, NSW 2006, Australia. Tel.: (61 2) 9036-9112; Fax: (61 2) 9351-3838; E-mail: yangpy@it.usyd.edu.au

Abstract

Ensemble learning is an intensively studies technique in machine learning and pattern recognition. Recent work in computational biology has seen an increasing use of ensemble learning methods due to their unique advantages in dealing with small sample size, high-dimensionality, and complexity data structures. The aim of this article is two-fold. First, it is to provide a review of the most widely used ensemble learning methods and their application in various bioinformatics problems, including the main topics of gene expression, mass spectrometry-based proteomics, gene-gene interaction identification from genome-wide association studies, and prediction of regulatory elements from DNA and protein sequences. Second, we try to identify and summarize future trends of ensemble methods in bioinformatics. Promising directions such as ensemble of support vector machine, meta-ensemble, and ensemble based feature selection are discussed.

Keywords: ensemble learning; bioinformatics; microarray; mass spectrometry-based proteomics; gene-gene interaction; regulatory elements prediction; ensemble of support vector machines; meta-ensemble; ensemble feature selection

1 INTRODUCTION

Modern biology has seen an increasing use of computational techniques for large scale and complex biological data analysis. Various computational techniques, especially machine learning algorithms [1], are applied, for example, to select genes or proteins associated with the trait of interest and to classify different types of samples in gene expression of microarrays data [2] or mass spectrometry (MS)-based proteomics data [3], to identify disease associated genes, gene-gene interactions, and gene-environmental interactions from genome wide association (GWA) studies [4], to recognize the regulatory elements in DNA or protein sequences [5], to identify protein-protein interactions [6], or to predict protein structure [7].

Ensemble learning is an effective technique that has increasingly been adopted to combine multiple learning algorithms to improve overall prediction accuracy [8]. These ensemble techniques have the advantage to alleviate the small sample size problem by averaging and incorporating over multiple classification models to reduce the potential for overfitting the *training data* [9]. In this way the training data set may be used in a more efficient way, which is critical to many biological applications with small sample size. Some ensemble methods such as *random forests* are particularly useful for high-dimensional datasets because increased classification accuracy can be achieved by generating multiple prediction models each with a different *feature* subset. These properties, as we will review later, have a major impact on many different bioinformatics applications.

*This manuscript has been published by Current Bioinformatics, 5, (4):296-308, 2010.

A large number of ensemble methods have been applied to biological data analysis. This article aims to provide a review of the most widely used methods and their variants used in bioinformatics applications, and to identify the future development directions of ensemble methods in bioinformatics. In the next section, we briefly discuss the rationale of ensemble approaches and introduce the three most popular ensemble methods – *bagging* [10], *boosting* [11], and *random forests* [12]. This is followed by a section discussing the application of ensemble methods to three different bioinformatics problems. These are: (1) gene expression of microarray and MS-based proteomics data classification, (2) identification of gene-gene interaction in GWA studies, and (3) regulatory elements prediction from DNA or protein sequences. Several other applications are also reviewed. The fourth section describes several extensions of ensemble methods and the adaptation of ensemble learning theory for feature selection problems. The last section concludes the paper.

2 POPULAR ENSEMBLE METHODS

Improvements in classification tasks are often obtained by aggregating a group of classifiers (referred to as *base classifiers*) as an ensemble committee and making the prediction for unseen data in a consensus way. The aim of designing/using ensemble methods is to achieve more accurate classification (on training data) as well as better generalization (on unseen data). However, this is often achieved at the expense of increased model complexity (decreased model interpretability) [13]. A better generalization property of ensemble approach is often explained using the classic bias-variance decomposition analysis [14]. Specifically, previous studies pointed out that methods like bagging (Fig. 1(a)) improve generalization by decreasing variance [15] while methods similar to boosting (Fig. 1(b)) achieve this by decreasing bias [16]. Here we provide a more intuitive interpretation of the advantage of ensemble approach.

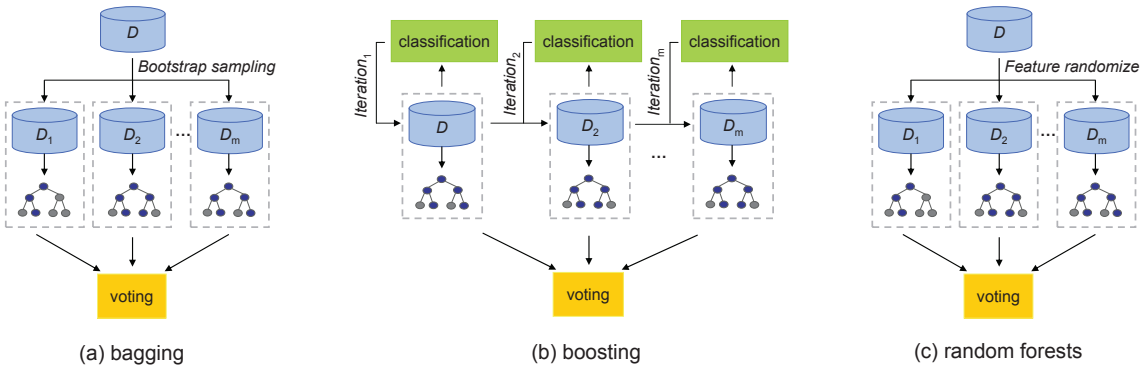
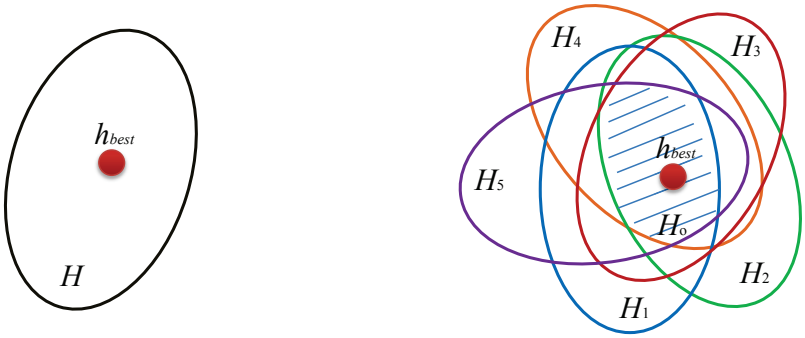


Fig. 1: Schematic illustration of the three popular ensemble methods.

Let the best classification rule (called *hypothesis*) h_{best} of a given induction algorithm for certain kind of data be the circle in Figure 2. Suppose the training data is free from noise, without any missing value, and sufficiently large to represent the underneath pattern. Then, we expect the classifier trained on the dataset to capture the best classification hypothesis represented as the circle. In practice, however, the training datasets are often compounded by small sample size, high dimensionality, and high noise-to-single ratio etc. Therefore, obtaining the best classification hypothesis is often nontrivial because there are a large number of suboptimal hypotheses in the hypothesis space (denoted as H in Figure 1) which can fit the training data but do not generalize well on unseen data.

Creating multiple classifiers by manipulating the training data in an intelligent way allows one to obtain a different hypothesis space with each classifier (H_1, H_2, ..., H_L; where L is the number of classifiers), which may lead to a narrowed overlap hypothesis space (H_o). By combining the classification rules of multiple classifiers using integration methods that take advantage of the overlapped region (such as averaging and majority voting), we are approaching the best classification rule by using multiple rules as an approximation. As a result, the ensemble composed in such a manner often appears to be more accurate.

From the above analysis, it is clear that in order to obtain an improvement the base classifiers need to be accurate (better than chance) and diverse from each other [17]. The need for diversity originates from the assumption that if a classifier makes a misclassification, there may be another classifier that



(a) Hypothesis space of a single classifier (b) Hypothesis space of an ensemble classifier

Fig. 2: A schematic illustration of hypothesis space partitioning with ensemble of classifiers. By combining moderate accurate base classifiers, we can approximate the best classification rule h_{best} with the increase of model complexity. This can be achieved by combining base classifiers with averaging or majority voting which takes advantage of the overlapped region.

complements it by correctly classifying the misclassified sample. Ideally, each classifier makes incorrect classification independently.

Popular ensemble methods like bagging (Fig. 1(a)) and random forests (Fig. 1(c)) (note that random forests can be considered as a special form of bagging algorithm) harness the diversity by using different perturbed data sets and different feature sets for training base classifiers, respectively. That is, each base classifier is trained on a subset of samples/features to obtain a slightly different classification hypothesis, and then combined to form the ensemble. As for boosting (Fig. 1(b)), diversity is obtained by increasing the weights of misclassified samples in an iterative manner. Each base classifier is trained and combined from the samples with different classification weights, and therefore, different hypotheses. By default, these three methods use decision trees as base classifiers because decision trees are sensitive to small changes on the training set [8], and thus suited for the perturbation procedure applied to the training data.

It is worth noting that there are many other well-established methods for creating ensemble classifier. For example, stacked generalization [18] combines the base classifiers through a meta-classifier to maximize the generalization. Methods like base classifier selection and cascade-based classifiers are also widely used [19, 20].

To aggregate the base classifiers in a consensus manner, strategies such as majority voting or simple averaging are commonly used. Assuming the prediction outputs of the base classifiers are independent of each other (which, in practice, is partially achieved by promoting diversity among the base classifiers), the majority voting error rate ϵ_{mv} can be expressed as follows [21]:

$$\epsilon_{mv} = \sum_{i=\lfloor M/2 \rfloor + 1}^{M} \binom{M}{i} \epsilon^i (1-\epsilon)^{M-i} \tag{1}$$

where M is the number of base classifiers in ensemble. Given the condition that $\epsilon < \epsilon_{random}$ for ϵ_{random} being the error rate of a random guess and all base classifiers have identical error rate ϵ, the majority voting error rate ϵ_{mv} monotonically decreases and approaches 0 when $M \to \infty$.

Figure 3 shows an ideal scenario in which the dataset has two classes each with the same number of samples, the prediction of base classifiers are independent of each other, and all base classifiers have identical error rate. It can be seen from the figure that, when the error rate of the base classifiers is smaller than 0.5, which is a random guess for a binary dataset with equal number of positive samples and negative samples, the ensemble error rate quickly gets smaller than the error rate of base classifiers. If we add more base classifiers, the improvement becomes more significant. In this example, we used odd numbers of base classifiers where the consensus is made by $(M+1)/2$ classifiers. When using even number of base classifiers, the consensus is made by $M/2+1$ classifiers.

Besides majority voting one can also apply other methods to combine base classifiers, such as weighted majority voting, bayesian combination [22], and probabilistic approximation [23]. Yet, majority vot-

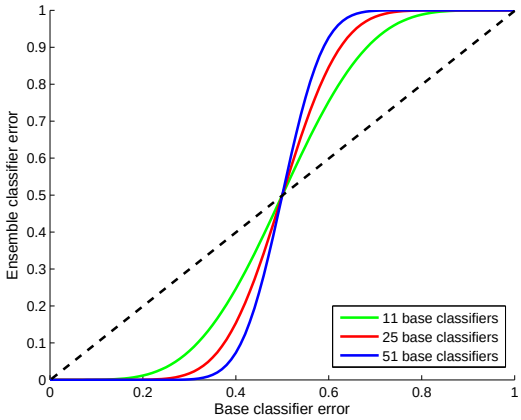


Fig. 3: Majority Voting. The relationship of error rates of base classifiers and error rates of the ensemble classifier in majority voting. The diagonal line represents the case in which the base classifiers are identical to each other, while the three carve lines represent combining different numbers of base classifiers which are independent of each other.

ing remains to be one of the most popular choices because of its simplicity and effectiveness compared to more complex decision fusion methods [24].

3 APPLICATION

In this section, we describe the application of ensemble methods in bioinformatics in three broad topics. They are as follow:

- Classification of gene expression microarray data and MS-based proteomics data;
- Gene-gene interaction identification using single nucleotide polymorphism (SNPs) data from G-WA studies;
- Prediction of regulatory elements from DNA and protein sequences

3.1 The application of ensemble methods to microarray and MS-based proteomics

Many biological studies are designed to distinguish patients from normal people, or to distinguish d-ifferent disease types, progression etc. based on the gene expression profiles or protein abundance. Typical high-throughput techniques include using microarray to measure gene expression [2] (Fig. 4) and using mass spectrometer to measure protein abundance [3] (Fig. 5). These techniques can provide a genome-wide transcription or translation monitoring. However, when such high-throughput techniques are applied, the experiments often result in the evaluation of a huge number of features (gene probsets in microarray studies or mass/charge (m/z) ratio in MS studies, etc.) with a limited number of samples [25]. This is commonly known as the "curse-of-dimensionality" [26], and selecting the most relevant features [27, 28] and making the most use of the limited data samples [29] are the key issues in microarray or MS-based proteomics classification problem.

The unique advantages offered by ensemble methods are their ability in dealing with small sample size and high dimensionality. For this reason, they have been widely applied to both microarray and MS-based proteomics data analysis.

The initial work of applying bagging and boosting methods to classify tumors using gene expression profiles was pioneered by Ben-Dor *et al.* [30] and Dudoit *et al.* [31]. Both studies compared the ensemble methods with other individual classifiers such as k-nearest neighbors (kNN), clustering based classifiers, support vector machines (SVM), linear discriminant analysis (LDA), and classification trees. The conclusion was that ensemble methods of bagging and boosting performed similarly to other single classification algorithms included in the comparison.

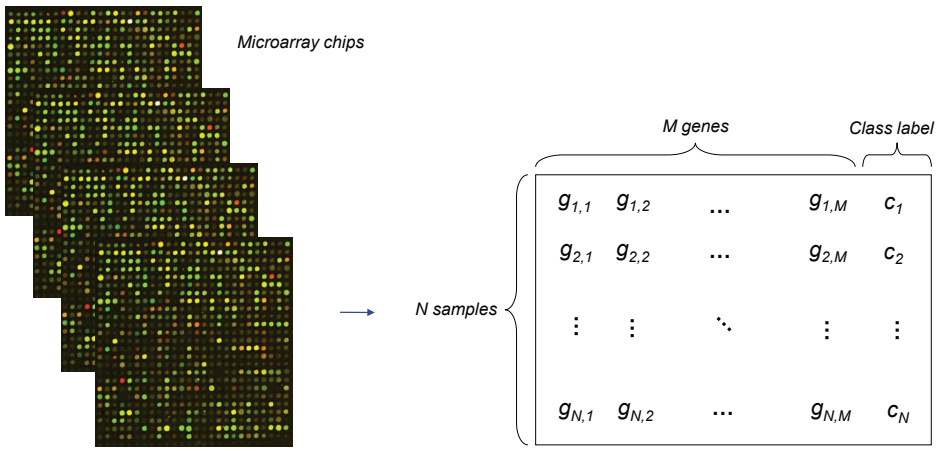


Fig. 4: Gene expression of microarray data matrix. The microarray data from the computational viewpoint is an $N \times M$ matrix. Each row represents a sample while each column represents a gene except the last column which represents the class label of each sample. $g_{i,j}$ is a numeric value representing the gene expression level of the i^{th} sample in the j^{th} gene. c_i in the last column is the class label of the i^{th} sample

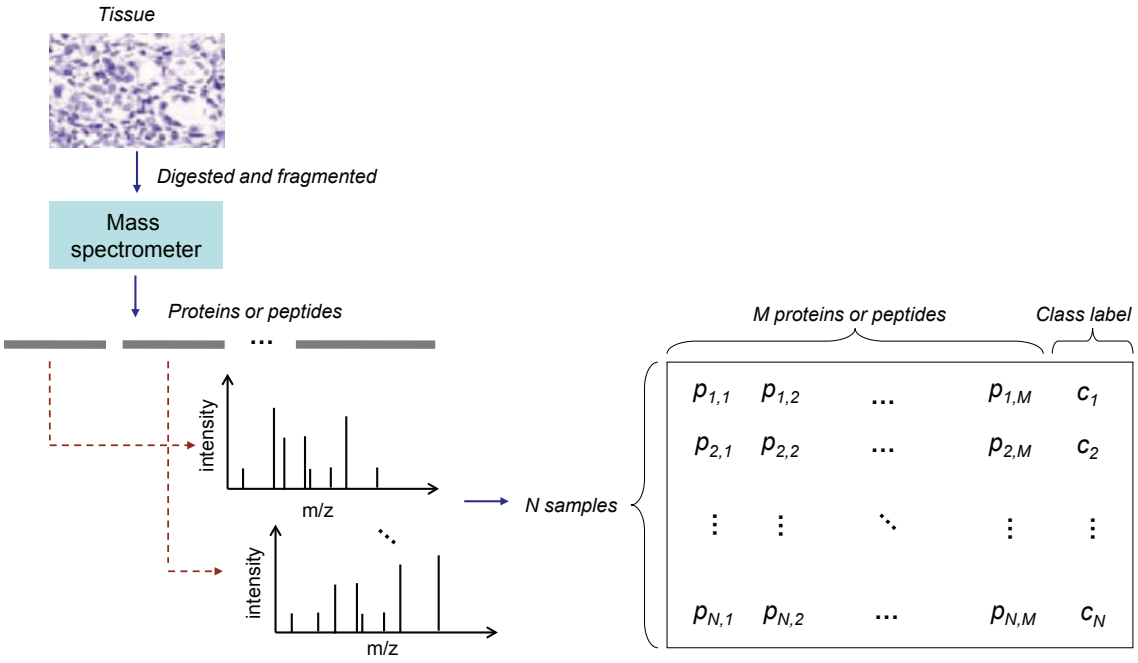


Fig. 5: Mass spectrometry-based proteomics. The proteomics data generated by mass spectrometer are very similar to microarray data from the computational viewpoint. The difference is that, instead of measuring gene expressions, each column represents the abundance of a protein or peptide in the tissue derived from a sample.

In contrast to the results obtained by Dudoit *et al.* and Ben-Dor *et al.*, the follow up studies revealed that much better results can be achieved through minor tuning and modification. For instances, Dettling and Bühlmann [32] proposed an algorithm called LogitBoost which replaces the exponential loss function used in AdaBoost with a log-likelihood loss function. They demonstrated that LogitBoost is

more accurate in classification of gene expression data compared to the original AdaBoost algorithm. Long [33] argued that the performance of AdaBoost can be enhanced by improving the base classifiers. He then proposed several customized boosting algorithms for microarray data classification. The experimental results indicate that the customized boosting algorithms performed favorably compared to SVM-based algorithms. In comparison to the single tree classifier, Tan and Gilbert [34] demonstrated that, overall, ensemble methods of bagging and boosting are more robust and accurate in microarray data classification using seven publicly available datasets.

In MS-based proteomics, Qu *et al.* [35] conducted the first study using boosting ensembles for classifying mass spectra serum profiles. 100% classification accuracy was estimated using the standard AdaBoost algorithm, while a simpler ensemble called boosted decision stump feature selection (BDSFS) showed slightly lower classification accuracy (97%) but gives more interpretable classification rules. A thorough comparison study was conducted by Wu *et al.* [36], who compared the ensemble methods of bagging, boosting, and random forests to individual classifiers of LDA, quadratic discriminant analysis, kNN, and SVM for MALDI-TOF (matrix assisted laser desorption/ionization with time-of-flight) data classification. The study found that among all methods random forests, on average, gives the lowest error rate with the smallest variance. Another recent study by Gertheiss and Tutz [37] designed a blockwise boosting algorithm to integrate feature selection and sample classification of mass spectrometry data. Based on LogitBoost, their method address the horizontal variability of the m/z values by dividing the m/z values into small subsets called blocks. Finally, the boosting ensemble has also be adopted as the classification and biomarker discovery component in the proteomic data analysis framework proposed by Yasui *et al.* [38].

In comparison to bagging and boosting ensemble methods, random forests holds a unique advantage because its use of multiple feature subsets is well suited for high-dimensional data such as those generated by microarray and MS-based proteomics studies. This is demonstrated by several studies such as [39] and [40]. In [39], Lee *et al.* compared the ensemble of bagging, boosting and random forests using the same experimental settings and found random forests was the most successful one. In [40], the experimental results through ten microarray datasets suggest that random forests are able to preserve predictive accuracy while yielding smaller gene sets compared to diagonal linear discriminant analysis (DLDA), kNN, SVM, shrunken centroides (SC), and kNN with feature selection. Other advantages of random forests such as robustness to noise, lack of dependence upon tuning parameters, and the speed of computation have been demonstrated by Izmirlian [41] in classifying SELDI-TOF proteomic data.

Due to the good performance of random forests in high-dimensional data classification, the development of random forests variants is a very active research topic. For instance, Zhang *et al.* [42] proposed a deterministic procedure to form a forest of classification trees. Their results indicate that the performance of the proposed deterministic forest is similar to that of random forests, but with better reproducibility and interpretability. Geurts *et al.* [43] proposed a tree ensemble method called "extra-trees" which selects at each node the best among k randomly generated splits. This method is an improvement on random forests because unlike random forests which are grown with multiple subsets, the base trees of extra-trees are grown from the complete learning set and by explicitly randomizing the cut-points.

Besides the development of more effective ensemble methods, current studies also focus on more objective comparison [44]. For example, a recent study by Ge and Wong [45] compared the single classifier of decision trees with six ensemble methods including random forests, stacked generalization, bagging, Adaboost, LogitBoost, and Multiboost using three different feature selection schemes (Student t-test, Wilcoxon rank sum test, and genetic algorithms). Another comprehensive study by Statnikov *et al.* [46] compared random forests with SVM for microarray-based cancer classification across 22 datasets.

Lastly, genes are connected by pathways and functioning in groups, and therefore, there is a growing trend to analyze microarray data at the pathway level [47]. Pang *et al.* [48] proposed to combining microarray data with the pathway information from the KEGG database [49]. The dataset is subsequently divided into categorical phenotype data and clinical outcome data, and then used to train a random forests ensemble. The genes selected by random forests for sample classification are treated as informative genes while the error rate of random forests is used to evaluate the association between pathway and the disease of interests.

3.2 The application of random forests to identify gene-gene interaction

Beside measuring gene expressions and protein expressions, screening and comparing the genotypes of different samples can also give critical information of different diseases and their pathogeneses because the development of the disease is studied from the very source of the genetic makeup — DNA. More

importantly, such studies, termed *association study*, can help to determine different individuals' susceptibility to various diseases as well as their response to different drugs based on their genetic variations [50].

A widely used design for association study is to screen common single nucleotide polymorphisms (SNPs) and compare the variation between case and control samples for disease associated gene identification at the genome-wide scale (termed as genome-wide association (GWA) studies) [4]. It is commonly accepted that many complex diseases such as diabetes and cancer arise from a combination of multiple genes which often regulate and interact with each other to produce the traits [51]. Therefore, the goal of these studies is to identify the complex interactions among multiple genes which together with environmental factors may substantially increase the risk of the development of diseases. Using SNPs as genetic markers, this problem is commonly formulated as the task of SNP-SNP and SNP-environment interaction identification. Figure 6 illustrates the pairwise interaction relationship among multiple SNPs.

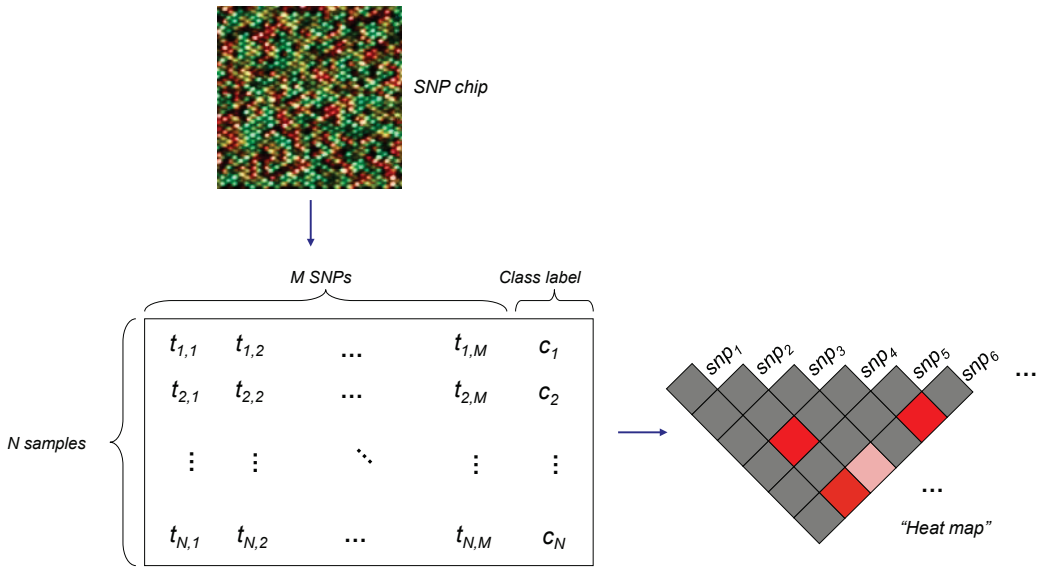


Fig. 6: Schematic illustration of SNP-SNP interactions. The SNP chip is applied for genotyping and the data matrix obtained is similar to those from microarray and MS-based proteomics studies except that each feature is a SNP variable which can take the genotype of *AA*, *AB*, or *BB*. The SNP-SNP interactions are schematically illustrated as the red boxes in the "heat map" with brighter colors indicating stronger interactions and associations with the disease of interest.

Among many pattern recognition algorithms, the decision tree algorithm has long been recognized as a promising tool for SNP-SNP interaction identification [52, 53]. Initial attempts to identify gene-gene interaction using decision tree based methods were investigated on relatively small datasets. For instance, Cook *et al.* [54] applied the CART algorithm with a multivariate adaptive regression spline model to explore the presence of genetic interactions from 92 SNPs.

With the increasing popularity of tree based ensemble methods, they became the focus of many recent studies under the context of SNP-SNP interaction identification for complex disease analysis. Although different ensemble methods have been proposed for identifying SNP-SNP interaction [55, 56], it is random forests that enjoyed the most popularity [51]. This is largely due to its intrinsic ability to take multiple SNPs jointly into consideration in a nonlinear fashion [57]. In addition, random forests can be used easily as an embedded feature evaluation algorithm [58], which is very useful for disease associated SNP selection.

The initial work of Bureau *et al.* [58] shows the advantage of random forests regression method in linkage data mapping. Several quantitative trait loci have been successfully identified. The same group [59] then applied the random forests algorithm in the context of the case-control association study. A similar method was also used by Lunetta *et al.* [60] for complex interaction identification. However, these early studies limited the SNPs under analysis to a relatively small number (30 - 40 SNPs).

Recent studies focus on developing customized random forests algorithms and applied them for gene-gene interaction identification to a much higher data dimension, containing several hundred thousands of candidate SNPs. Specifically, Cheng *et al.* [61] investigated the statistical power of random

forests in SNP interaction pair identification. Their algorithm was then applied to analyze the SNP data from the complex disease of age-related macular degeneration (AMD) [62] by using a haplotype based method for dimension reduction. Meng *et al.* [63] modified random forests to take into account the linkage disequilibrium (LD) information when measuring the importance of SNPs. Jiang *et al.* [64] developed a sequential forward feature selection procedure to improve random forests in epistatic interaction identification. The random forests algorithm was first used to compute the *gini index* for a total of 116,204 SNPs from the AMD dataset [62] and then used as a classifier to minimize the classification error by selecting a subset of SNPs in a forward sequential manner with a predefined window size.

3.3 The application of ensemble methods to regulatory elements prediction

Regulatory elements prediction is a general term that encompasses tasks such as promoter region recognition [5, 65], transcription start sites prediction [66], or glycosylation site and phosphorylation site prediction [67]. The similarity of these tasks is to computationally identify the functional sites based on the sequences of DNA or proteins with other biological and/or genomic information. Figure 7 illustrates different functional sites on a DNA sequence of a gene.

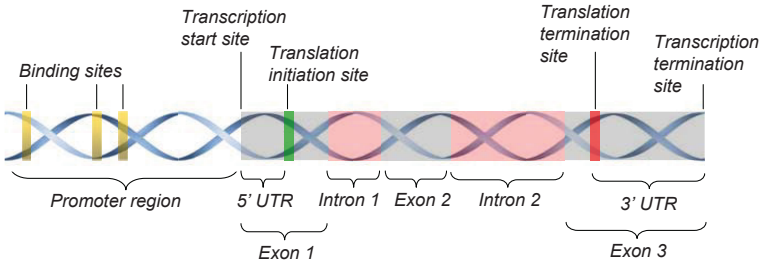


Fig. 7: Schematic illustration of functional sites. This is a schematic illustration of different functional sites on a DNA sequence of a gene. The task of regulatory elements prediction could be the computational identification of the promoter region–promoter region recognition, or the computational identification of the transcription start sites–transcription start sites recognition, etc.

Ensemble methods have recently been introduced to this domain due to the diverse data types and features each task employs to perform the recognition and the diverse patterns presented in different promoter sequences. For instance, Hong *et al.* [68] proposed a modified boosting approach for identifying transcription factor binding sites. The modified boosting algorithm was applied to ChIP-chip data. It automatically decides the number of base classifiers to be used so as to avoid overfitting. Xie *et al.* [69] utilized the AdaBoost algorithm to combine a variety of features for promoter site identification. The features included ranges from local distribution of pentamers, positional CpG island (genomic regions with high CpG sites) features, to digitized DNA sequence. AdaBoot is adopted to select the most informative features while building the ensemble of classifiers. Zhao *et al.* [70] adopted a similar method that utilizes LogitBoost with stumps for transcription start sites prediction. They used a diverse collection of features including core promoter elements, transcription factor binding site, mechanical properties, markovian score, and k-mer frequency. The resulting program called CoreBoost contains two classifiers which are specific for CpG-related promoter prediction and for non-CpG-related promoter prediction, respectively. By integrating specific genome-wide histone modification as a set of extra features, Wang *et al.* [71] proposed an improved CoreBoost algorithm called CoreBoost with histone modification features or CoreBoost_HM. They then demonstrated that CoreBoost_HM can successfully used to predict of core-promoters of both coding and noncoding genes. Quite uniquely, Gordon *et al.* [72] combined a group of SVMs, each with a different mismatch string kernel, for transcription start sites prediction. They found a significantly reduced false positives in the prediction result which, from a practical viewpoint, is extremely useful to biologists.

Glycosylation site and phosphorylation site are the functional sites of post translational modifications (PTMs) in protein sequences. Accurate localization of these functional sites can elucidate many important biological process such as protein folding, subcellular localization, protein transportation and functions. In [73], Hamby and Hirst utilized the random forests algorithm for glycosylation sites prediction and prediction rule extraction. The significant increase of prediction accuracy is observed in the prediction of Thr and Asn glycosylation sites. In [74], Caragea *et al.* attempted to devise an ensem-

ble using bagging with the base classifier of SVMs. Their comparison to single SVM indicates that the ensemble of SVM is more accurate according to several evaluation metrics.

Moreover, ensemble methods can be used as an embedded component for model tuning. A typical example is the study [75] in which the Yoo *et al.* employed the AdaBoost algorithm for tuning multiple neural networks. The tuned system was then used for phosphorylation site prediction, and the performance of the this system compared favorably to nine existing machine learning algorithms and four widely used phosphorylation site predictors.

3.4 Other emerging applications of ensemble methods in bioinformatics

Besides the above three main areas, ensemble methods have also been widely applied to many other different bioinformatics problems.

In gene function prediction, Guan *et al.* [76] introduced a meta-ensemble based on SVM. This meta-ensemble contains three "base classifiers". They are the ensemble of SVMs trained using bagging for each gene ontology (GO) term, the hierarchical bayesian combination of SVM classifiers, and the naïve bayes combination of SVM classifiers. The prediction of this meta-ensemble is made by selecting the best performing one on each GO term.

Protein folding recognition, structure prediction, and function prediction are closely related problems. In [77], Shen and Chou designed nine sets of features for ensemble recognition of protein folding. The features extracted from the protein sequences include predicted secondary structure, hydrophobicity, van der Waals volume, polarity, polarizability, and different dimensions of pesudo-amino acid composition. Modified kNN base classifiers are trained using different feature sets and combined in a weighted voting manner.

Melvin *et al.* [78] proposed to combine kNN classifier with SVM classifier for protein structure prediction using sequence information. The kNN classifier is trained using global sequence information (called full-coverage) while the SVM is trained using local sequence information. The classifiers are then combined by a punting method using a specified threshold. Lee *et al.* [79] compare random forests to SVM in identifying protein functions with features derived from protein sequence properties. In their study, 484 features are extracted solely from the protein sequence and a correlation-based feature selection (CFS) procedure coupled with either SVM (SVM_CFS) or random forests (RF_CFS) is applied to identify the final 39 features which is then used in function classification of 11 different protein classes. The performance of SVM_CFS and RF_CFS are compared with SVM and RF without feature selection using five evaluation metrics. Overall, SVM and RF are comparable, and when coupled with CFS the performance can be significantly improved. Wang *et al.* [80] employed stacked generalization to predict membrane protein types. A SVM and a kNN were used as the base classifiers and a decision tree was adopted to combine the base classifiers.

The problem of protein-protein interaction prediction has also been approached from the ensemble perspective. In study [81], Chen and Liu introduced a domain-based random forests method to infer protein interactions. The protein-protein interactions are inferred from the protein domain level, and the proposed "domain-based random decision forest framework" predicts possible domain-domain interactions by considering all single-domains as well as domain combination pairs. More recently, Deng *et al.* [82] applied a SVM-based ensemble algorithm using booststrap resampling and weighted voting strategy for protein-protein interaction sites prediction. One difficulty of this learning task is the imbalanced of the data classes due to the lack of positive training examples. Deng *et al.* found that their ensemble of SVMs can alleviate the imbalanced problem and significantly improve the prediction performance.

Finally, many recently studies also focus on elucidating genetic networks using ensemble methods. For instance, Wu *et al.* [83] proposed to use a relevance vector machine (RVM)-based ensemble for prediction of human functional genetic networks from multiple sources of data. The proposed ensemble is combined in a boosting manner and the comparison with a naïve bayes classifier indicate that the ensemble is more effective even with massive missing values. The study of Altay and Emmert-Streib [84] adopting ensemble approach from a different perspective. In particular, they use an ensemble of datasets drawn under the same condition to reveal the differences in gene networks inferred by different algorithms. They identified the bias of different inference algorithms with respect to different network components, and subsequently, use this information to interpret more objectively on the inferred networks.

The applications of ensemble methods in bioinformatics reviewed above are by no means an exhaustive list but merely the major topics which have received much attention. In most reviewed studies,

ensemble methods have shown to be very useful. Given the flexibility and the numerous ways to create and tune them, it is likely that much more effort will be directed to solve many more biological problems (both old and new) from the ensemble perspective in the coming years.

4 EXTENSION OF ENSEMBLE METHOD IN BIOINFORMATICS

The accumulating evidence suggests that the ensemble method is one of the most promising solutions to many biological problems. Due to the immense success of many ensemble methods in bioinformatics applications, numerous extensions have been proposed. In this section, we review some of the most promising directions. They are divided into two major topics. The first one discusses different extensions for achieving better prediction, while the second one discusses the adaptation of the ensemble theory for feature selection – ensemble feature selection.

4.1 The extension of ensemble methods for classification

4.1.1 Ensemble of SVMs

SVM is generally considered as the best "off-the-shelf" classifier. If it can be successfully used as the base classifier of an ensemble, the further improvement gain could be noteworthy. One simple way to use SVM in the ensemble framework is to apply bagging procedure with the base classifier of SVM. This is the approach taken by Caragea *et al.* [85] who applied a bagging ensemble with the base classifier of SVM for glycosylation site prediction. The experimental results indicate that by training each base classifier with a re-sampling of the "balanced" training set, the performance of the SVM ensemble suppresses both the single SVM and the balanced SVM. Similarly, in [76], Guan *et al.* applied the bagging procedure for constructing an ensemble of SVMs for gene function prediction. In gene ontology (GO) term recognition, the ensemble of SVMs consistently outperformed the single SVM classifier.

In the study of Peng [86], the concept of over-generating and selecting of an appropriate subset of base classifiers were investigated. The base classifier used is SVM and the bootstrap sampling method is used to generate multiple training sets. Compared to the decision tree, SVM is much more stable for small perturbation of the training samples. In order to obtain the diversity among the base classifiers, a clustering based base classifier selection procedure is employed to explicitly ensure that the base classifiers are accurate while also disagreeing with each other. By comparing it to a single SVM classifier and the ensemble of bagging and boosting, Peng demonstrated that the proposed clustering based SVM ensemble achieved the best result.

The study by Gordon *et al.* [72] utilized a unique ensemble approach in which multiple SVM each with a different kernel is combined for transcription start sites prediction. This approach provides a new way to create ensemble of SVMs. It could be extremely useful to the problems with heterogenous data sources and feature types.

4.1.2 Meta ensemble

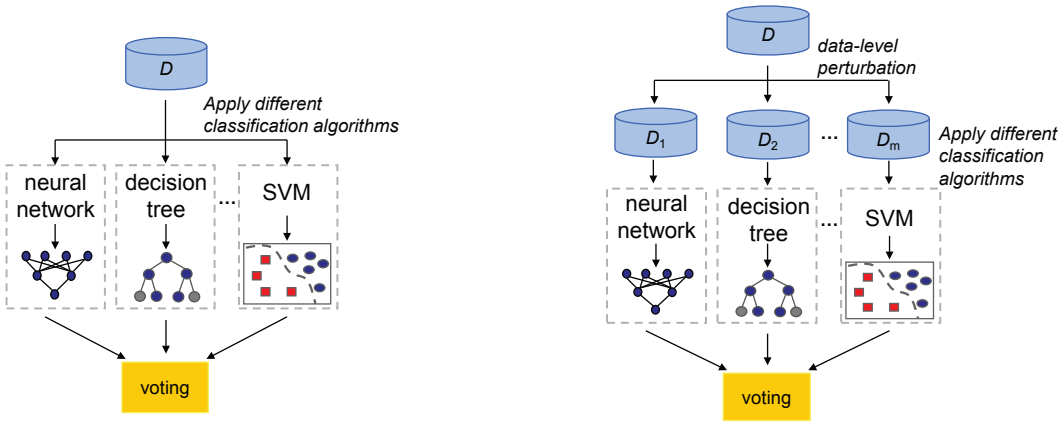
One pursuable idea is to gain more improvement by building the ensemble of ensembles – meta ensemble. This idea was first investigated by Dettling [87] who proposed to combine the bagging and boosting algorithms (called BagBoosting) for microarray data classification. The underlying hypothesis is that the boosting ensemble has a lower bias but the variance is relatively high, while the bagging ensemble has a lower variance but approximately non-altered bias. Therefore, combining these two ensemble methods may result in a prediction tool which could achieve both low bias and low variance. The empirical evaluation indicates that the proposed BagBoosting can improve the predication compared to bagging and boosting alone, and it is competitive compared to several other classifiers such as SVM, kNN, DLDA, and PAM. In [76], three different ensembles of SVMs are treated as "base" classifiers and are further combined as a meta-ensemble of SVMs for gene function prediction. The final prediction of genes are made by selecting the best performing classifier according to each GO term. Another study by Liu and Xu [88] explored a different way of forming meta-ensemble of classifiers. Their ensemble system is based on a genetic programming approach which optimizes a group of small-scale ensembles, called sub-ensembles, each consisting of a group of decision trees trained using different sets of input features. The experiment demonstrates that the system outperforms several other evolutionary based algorithms.

4.1.3 Ensemble of multiple classification algorithms

Another direction for extending the ensemble idea is to gain the disagreement in sample classification by using different classification algorithms. That is, instead of manipulating the dataset to train different classification models using a given classification algorithm such as decision trees or SVM, these methods attempt to find the diversity of the base classifier by using heterogenous classification algorithms.

For example, Bhanot *et al.* [89] combined ANN, SVM, Weighted Voting, kNN, decision trees, and logistic regression for the classification of mass spectrometry data. Kedarisetti *et al.* [90] extracted different sets of features from the protein sequence database to train an ensemble of classifiers using kNN, decision trees, logistic regression, and SVM classification algorithms. The ensemble is then used for protein structural class prediction. Hassan *et al.* [91] combined a set of fifteen classifiers ranging from rule-based classifiers such as kNN and decision trees to function-based classifiers such as SVM and neural networks. This ensemble of classification algorithms is applied to three microarray datasets to find a small number of highly differentially expressed (DE) genes. Yang *et al.* [92] proposed multi-filter enhanced genetic ensemble system for microarray analysis. The system combines multiple classifiers and filtering algorithms with a multiple objective genetic algorithm. By introducing a combinatorial ranking component and optimizing a set of base classifiers, Yang *et al.* [93] extend the genetic ensemble system for gene-gene interaction identification from GWA studies.

The similarity of this class of ensemble methods is that the diversity of the ensemble classifier is imposed by using different classification algorithms. However, this could be further combined with data-level perturbation to produce a meta-ensemble of classifiers, which could potentially increase the overall diversity while providing higher classification accuracy. The schematic illustration of such kind of ensemble methods is depicted in Figure 8.



(a) without data perturbation

(b) with data perturbation

Fig. 8: Schematic illustration of the ensemble using different classification algorithms. (a) classification algorithms are trained using the same training set. (b) classification algorithms are trained using different perturbations of the training set.

4.1.4 Other approaches

It is possible to create ensembles using many other approaches. Liu *et al.* [94] introduced a novel ensemble of neural networks by using three different feature selection/extraction methods coupled with bootstrapping to generate diverse base classifiers. Their study demonstrated that the diversity of base classifiers can also be obtained by incorporating different feature generating algorithms which provide several different gene ranking lists. A similar idea was used by Koziol *et al.* [95]. They assembled five disjoint lists of genes and created the base classifiers of decision trees, each trained on the dataset filtered by a gene list. The prediction is then made by simple voting.

To enhance the random forests algorithm for very high-dimensional dataset, Amaratunga *et al.* [96] designed a random forests variant called enriched random forests which weigh the importance of features when selecting splitting nodes. This modified random forests demonstrated very promising results

when the dimension of the microarray data is huge while the number of the discriminative genes is small. More selection chance are given to these informative genes while the diversity of the base classifier is still preserved by also including other different genes for building the base classifier.

Yanover *et al.* [97] introduced a statistical ensemble method called solution-aggregating motif finder (SAMF). Their method is based on Markov Random Field with the BMMF algorithm [98] which gives the M top-scoring solutions. The final result is given by aggregating the clustering output of the BMMF solutions.

There are also many extensions of ensemble methods under the Bayesian framework. For example, Armañanzas *et al.* [99] proposed a hierarchy of Bayesian network classifiers for detecting gene interactions from microarray data, and Robles *et al.* [100] used Bayesian network to combine multiple classifiers for protein secondary structure prediction.

Finally, utilizing the general theory of ensemble, Hu *et al.* [101] and Wijaya *et al.* [102] proposed to combining the outputs of multiple motif finder algorithms so as to improve the final prediction result. In this regard, the focus is shifted to the design of proper integration function for combining multiple results.

4.2 The adaptation of the ensemble theory for feature selection

The idea of ensemble in biological data analysis originates from combining multiple classifiers for improving sample classification accuracy. However, it has been adapted and increasingly used in feature selection, possibly as a consequence of the growing concern of the instability of the feature selection results from high-dimensional data [103].

One direct adaptation of ensemble methods for feature selection is to modify them as embedded feature selection algorithms by incorporating a feature extraction component. This idea is very similar to the use of random forests for SNP-SNP interaction identification. For instance, Jiang *et al.* [104] employed a gene shaving method with random forests so as to select differentially expressed genes. Levner [105] designed a feature extraction procedure for a boosting ensemble. The feature selection procedure is similar to sequential forward selection procedure in that the algorithm selects a single best feature during each boosting iteration. Saeys *et al.* [106] also applied random forests as an embedded feature selection algorithm. However, it is further combined with two other filtering algorithm – Symmetrical Uncertainty and RELIEF, and an SVM with recursive feature elimination (SVM-RFE). The results of these studies generally support the adaptation of the ensemble method for feature selection.

A more general approach is to utilize the ensemble theory of combining multiple models. Specifically, Dutkowski and Gambin [107] combined several filtering algorithms in a cross-validation framework for biomarker selection from mass spectrometry data. Multiple classification algorithms are used to evaluate the selected biomarkers so as to yield more stable results. Zhang *et al.* [108] incorporated multiple filtering algorithms and classification algorithms to improve the prediction accuracy and the stability of the gene ranking results in a genetic algorithm based wrapper procedure. Abeel *et al.* [109] studied the ensemble of filters in a bootstrap framework. Netzer *et al.* [110] developed a feature selection approach using the principle of stacked generalization. The feature selection algorithm termed stacked feature ranking is reported to identify important markers and improve sample classification accuracy.

Yang *et al.* [111] integrated various statistical methods such as t-test, penalized t-test, mixture models, and linear models to improve the robustness of the gene ranking results of microarray. Similarly, Chan *et al.* [112] combined Wilcoxon test with different feature selection procedures and different classification algorithms. They divided the feature selection into two levels–statistical feature selection and secondary feature selection. The underlying principle behind these methods is that genes and proteins that are selected or highly ranked by different measures are more likely to have genuine biological relevance than those by a single measure [111].

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

Feature selection is a key technique originated from the fields of artificial intelligence and machine learning [113, 114] in which the main motivation has been to improve sample classification accuracy [115]. 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 [116], for example, to identify potential disease associated genes from microarray studies [117], proteins from *mass spectrometry* (MS)-based proteomics studies [105], or *single nucleotide polymorphism* (SNP) from *genome wide association* (GWA) studies [118]. 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 [119]. 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.

6 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 [103, 120]. 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.

6.1 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 [114]. Figure 9 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 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 [121]. 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 [122].

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 [123]. 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 can-

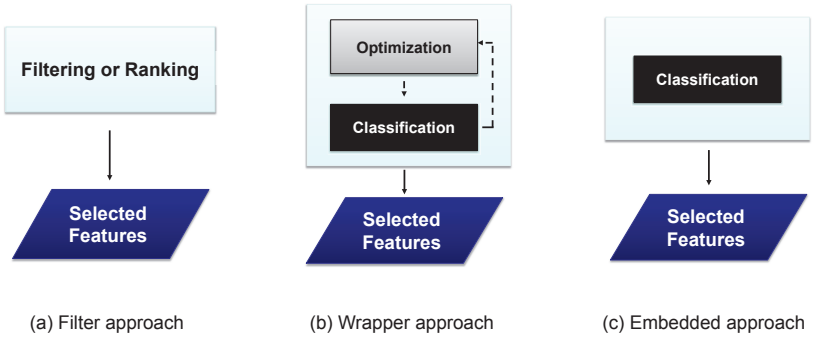


Fig. 9: Categorization of feature selection algorithms. (a) Filter approach where feature selection is independent from the classification. (b) Wrapper approach where feature selection relies on an inductive algorithm for sample classification in an iterative manner. (c) Embedded approach where feature selection is performed implicitly by an inductive algorithm during sample classification.

didates. 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 [124, 125, 126], and greedy approaches such as incremental forward selection [127], and incremental backward elimination [128].

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) [129] and random forest based feature evaluation [130].

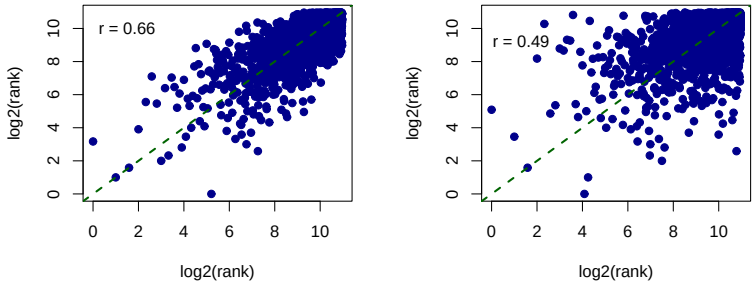
6.2 Potential causes of feature selection instability

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 [109]. 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 [131]. 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 [132].

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.

6.2.1 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 [133]. 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

Fig. 10: Instability demonstration for a filter algorithm (moderated t-test) and an embedded algorithm (SVM-RFE) in feature selection on colon cancer microarray dataset [133]. (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 10a shows the scatter plot of two runs of a filter algorithm known as moderated t-test [121]. 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 10b 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.

6.2.2 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 [134] 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 $j = 1 \dots N$ and $i = 1 \dots M$. *ReliefF* algorithm ranks each SNP, by updating a weight function for each SNP at each iteration as follows:

$$W(g_j) = W(g_j) - D(g_j, s_i, h_k)/M + D(g_j, s_i, m_k)/M \tag{2}$$

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:

$$D(g, s_a, s_b) = \begin{cases} 0 & \text{if } G(g, s_a) = G(g, s_b) \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

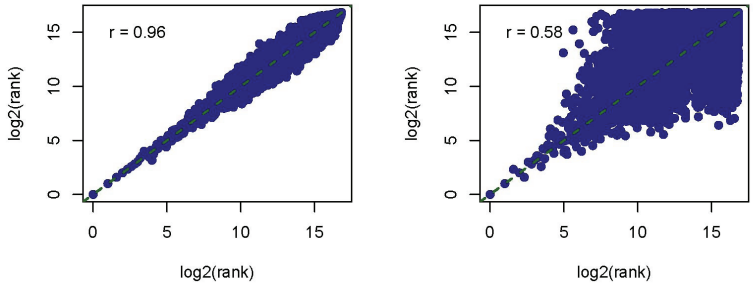
where $G(.)$ denotes the genotype of SNP g for sample s. The nearest neighbors to a sample are determined by the distance function, $MD(.)$, between the pairs of samples (denoted as s_a and s_b) which is also based on the difference function (Eq. 3):

$$MD(s_a, s_b) = \sum_{j=1}^{N} D(g_j, s_a, s_b) \quad (4)$$

Turned ReliefF (TuRF) proposed by Moore and White [122] 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 [62]. 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 11a,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

Fig. 11: Instability demonstration for *ReliefF* and TuRF algorithms. (a) scatter plot of two runs of *ReliefF* one on the original AMD dataset [62] 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.

6.2.3 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.

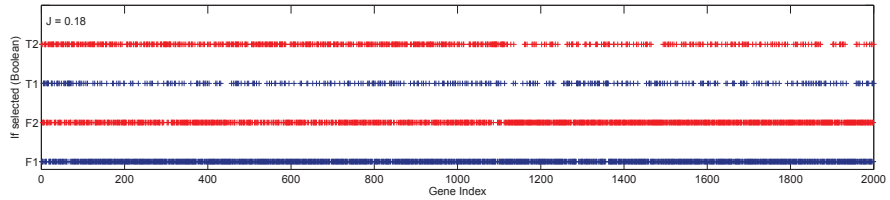


Fig. 12: Instability demonstration for GA/kNN wrapper algorithm with colon cancer microarray dataset. The x-axis is the index of the 2000 genes in the colon cancer microarray dataset [133]. The y-axis is a boolean value indicating whether a gene is selected. Two separate runs of GA/kNN each with a different 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 8.2 for details).

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.* [124], 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 [135]. Figure 12 shows two separate runs of GA/kNN wrapper algorithm each with a different 5-fold cross validation partitioning of colon microarray dataset [133] for model training and feature evaluation. After running GA/kNN, a gene is either selected or 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 8.2 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.

6.3 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 7. A review for several other approaches can be found in [136].

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 8.

7 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 [109, 137, 138]. 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.

7.1 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 [103, 109, 132]. The idea is built on the successful experience in ensemble classification [139] 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 13.

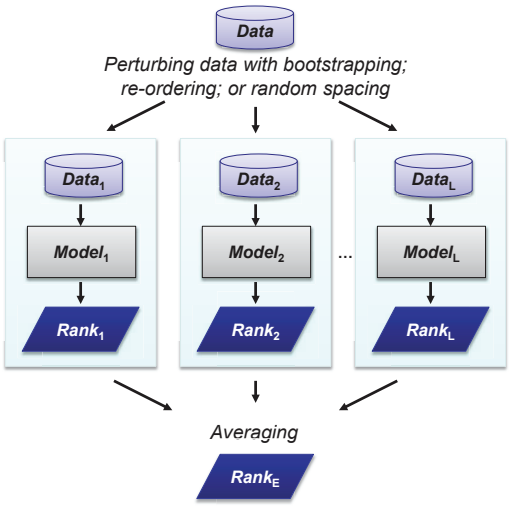


Fig. 13: Schematic illustration of an ensemble of filters using data perturbation approach.

7.2 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 14. This methods is firstly described in [131] where a *forward feature selection* (FFS) wrapper and a *backward feature elimination* (BFE) wrapper are shown to benefit from this ensemble approach.

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.

7.3 Performance on feature selection stability

We continue the examples in Section 6.2 and evaluate the performance of ensemble feature selection algorithms in terms of feature selection stability.

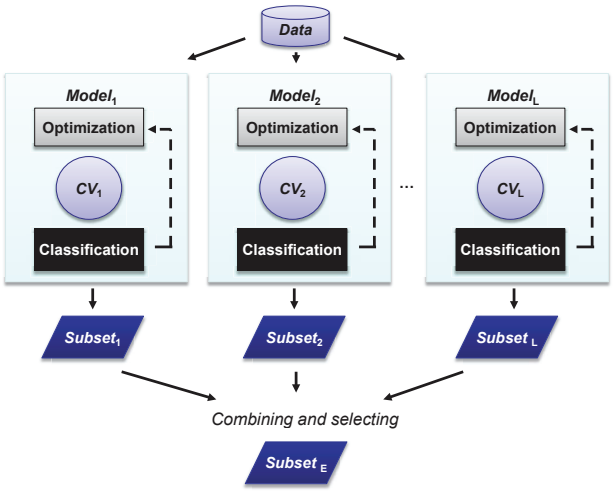
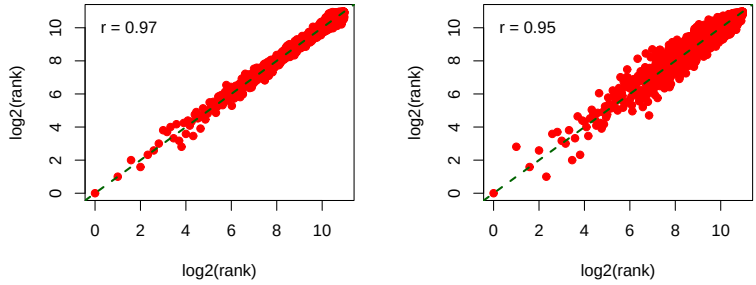


Fig. 14: Schematic illustration of an ensemble of wrappers using different partitions of an internal cross validation for feature evaluation.

7.3.1 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 15a,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 [133]. 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.



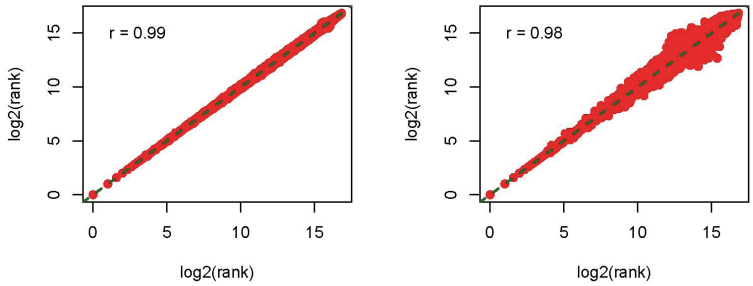
(a) Ensemble of moderated t-test (b) Ensemble of SVM-RFE

Fig. 15: 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 [133]; (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 10a). Similarly, the ensemble of SVM-RFE is able to increase the Spearman correlation from 0.49 to 0.95.

7.3.2 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 16 shows the ensemble version of *ReliefF* and TuRF algorithms where an ensemble size of 50 is used.



(a) Ensemble of *ReliefF* (b) Ensemble of TuRF

Fig. 16: 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 [62]; (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.

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 11b) whereas the ensembles of TuRF improve the Spearman correlation to 0.98 (Figure 16b).

7.3.3 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 [133]. Figure 17 shows the concordance of two ensemble of GA/kNN. The Jaccard set-based index increases from 0.18 (Figure 12) 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.

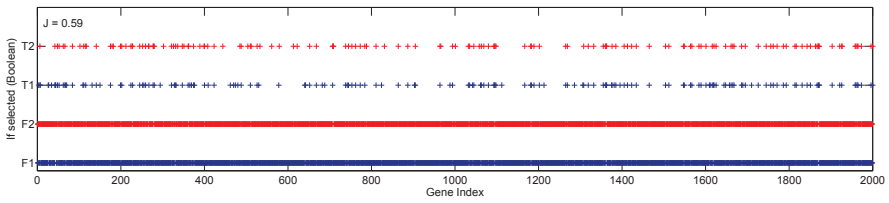
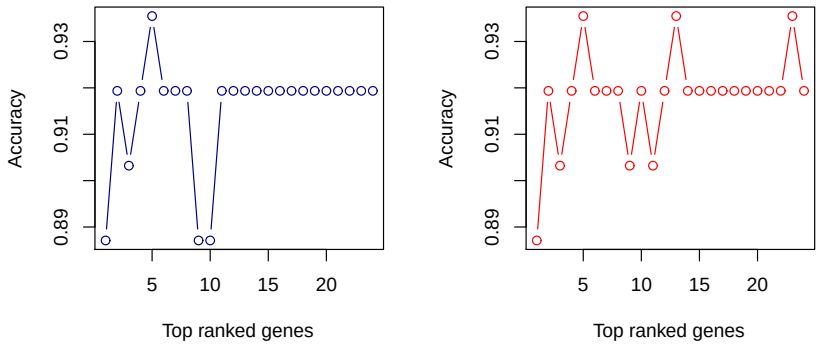


Fig. 17: Ensemble feature selection algorithms for data partitioning. The x-axis is the index of the 2000 genes in the colon cancer microarray dataset [133]. The y-axis is a boolean value indicating whether a gene is selected. Two separate runs of ensemble of GA/kNN (an ensemble of 50) each with a different 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 8.2 for details).

7.4 Performance on sample classification

Besides improving stability, another goal is to achieve higher classification accuracy by using ensemble feature selection approach [109]. Here we tested the classification accuracy using the genes selected

by moderated t-test from colon cancer microarray dataset [133], 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

Fig. 18: Sample classification accuracy using genes selected from colon cancer microarray dataset [133] by (a) moderated t-test, and (b) ensemble of moderated t-test.

From Figure 18, 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.

7.5 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 [133]. Several different evaluation metrics are used to assess the stability. Those metrics are described in details in Section 8.

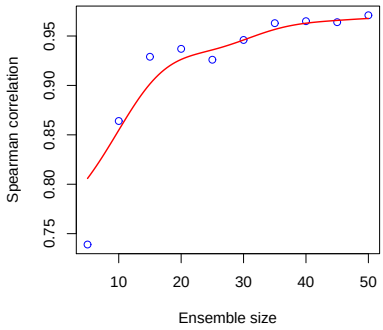
From Figure 19a,b, we can see that larger ensemble size of the moderate t-test corresponds to higher feature selection stability in terms of both Spearman correlation and Jaccard rank-based index. Similar effect is also observed for the ensemble of GA/kNN where increasing ensemble size results in higher feature selection stability as indicated by Jaccard set-based index (Figure 19c).

The size of the selected gene subsets has been used by many studies as an evaluation standard for wrapper algorithms [127]. 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 19d.

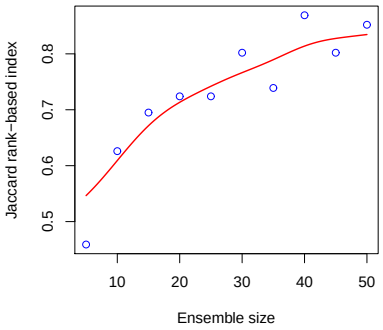
7.6 Some key aspects in ensemble feature selection algorithms

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.

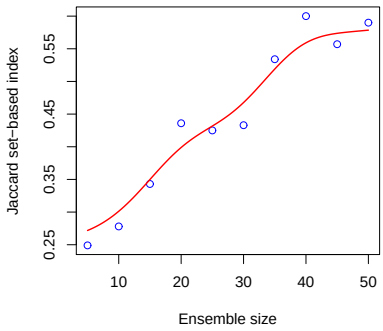
The ensemble approach attempts to improve feature selection result by increasing model complexity. Why the added complexity may improve the feature selection result leads to the second key aspect known as diversity which is intensively studied in designing ensemble of classification algorithms [140].



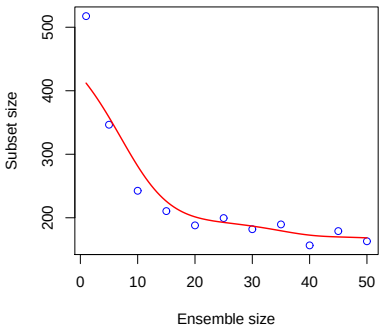
(a) Ensemble of moderated t-test (Spearman)



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



(c) Ensemble of GA/kNN (Jaccard)



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

Fig. 19: 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.

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 [141] 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.

8 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 [131]. Formally, this

procedure can be expressed as follows:

$$\overline{S}_p = \frac{2}{L(L-1)} \sum_{i=1}^{L} \sum_{j=i+1}^{L} S(\mathcal{F}_i, \mathcal{F}_j) \tag{5}$$

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.

8.1 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 8.2.

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.

8.1.1 Full ranking metrics

The most widely used metric for full ranking list is probably *Spearman correlation* [119, 142, 106]. For stability assessment, it is applied as follows:

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

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.

8.1.2 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 [143]. For example, the *Jaccard rank-based index* is a typical partial ranking metric used for assessing stability of feature selection algorithm in several studies [119, 106]. 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:

$$S_J^k(\mathcal{R}_i, \mathcal{R}_j) = \sum_{\tau=1}^{N} \frac{I(r_i^\tau \leqslant k \wedge r_j^\tau \leqslant k)}{2k - I(r_i^\tau \leqslant k \wedge r_j^\tau \leqslant k)} \tag{7}$$

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 [144]:

$$S_I^k(\mathcal{R}_i, \mathcal{R}_j) = \sum_{\tau=1}^{N} I(r_i^\tau \leqslant k \wedge r_j^\tau \leqslant k) \frac{1 - k^2/N}{k - k^2/N} \tag{8}$$

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.

8.2 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.* [131] 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:

$$S_H(\mathcal{M}_i, \mathcal{M}_j) = 1 - \sum_{\tau=1}^{N} \frac{|m_i^\tau - m_j^\tau|}{N} \tag{9}$$

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:

$$S_J(\mathcal{M}_i, \mathcal{M}_j) = \sum_{\tau=1}^{N} \frac{m_i^\tau \wedge m_j^\tau}{m_i^\tau \vee m_j^\tau} \tag{10}$$

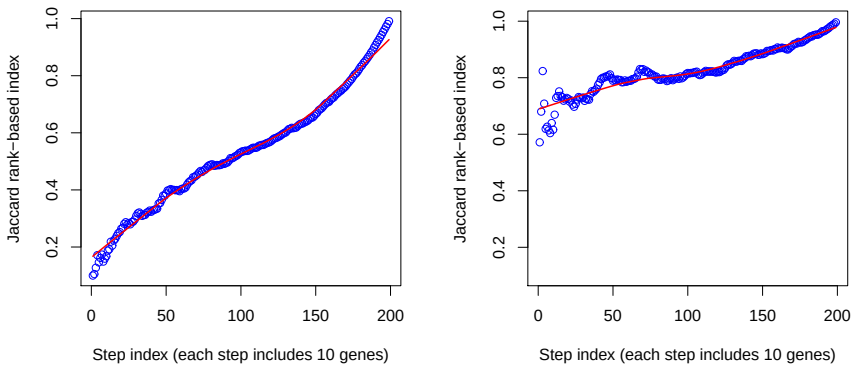
where m_i^τ and m_j^τ as before denote the value of the τth position in the boolean list of i and boolean list of j. The term $m_i^\tau \wedge m_j^\tau$ over the sum of total number of features N gives the intersection of selected features in the two boolean list, whereas the term $m_i^\tau \vee m_j^\tau$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

8.3 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 20 shows the stability evaluation across multiple thresholds of Jaccard rank-based index. In particular, Figure 20a is the result using SVM-RFE and Figure 20b is the result using ensemble of SVM-RFE all with colon cancer microarray dataset [133]. 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

Fig. 20: Evaluation of using different thresholds for computing stability. SVM-RFE and its ensemble are run on bootstraps on colon cancer microarray dataset [133], 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.

8.4 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 [119, 143]. 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.

9 CONCLUSIONS

In classification and prediction, a carefully engineered ensemble algorithm generally offer higher accuracy and stability than a single algorithm can achieve. In addition, ensemble algorithms can often alleviate the problems of small sample size and high dimensionality which commonly occur in many bioinformatics applications. It is worth mentioning that the increased accuracy is often accompanied

with increased model complexity which causes decreased model interpretability and higher computational intensity. Nevertheless, the theoretical studies of ensemble approaches and the increase of computational power may counter those difficulties.

Beside classification, many ensemble methods can also be used, with minor modifications, for feature selection or measuring feature importance. These are the main tasks in many biological studies such as disease associated gene selection from microarray, disease associated protein selection from mass spectrometry data, or high risk SNPs and SNP-SNP interaction identification from GWA studies. In feature selection, the development of novel methods which are guided by general ensemble learning theory has been proved to be fruitful. Therefore, they are likely to be effective methods and tools to address the ever-widening gap between the sample size and the data dimension generated by high-throughput biological experiments.

This review mainly focused on the most popular methods and the main applications. Yet, the idea of ensemble has been widely applied to many other bioinformatics problems, which is beyond the scope of this review. The utilization of ensemble methods has been one of the recent growing trends in the field of bioinformatics. It is our expectation that ensemble methods will become a flexible and promising technique for addressing many more bioinformatics problems in the years to come.

10 SUMMARY

- Ensemble methods have been increasingly applied to bioinformatics problems in dealing with small sample size, high-dimensionality, and complexity data structure.
- The main applications of ensemble methods in bioinformatics are classification of gene expression and mass spectrometry-based proteomics data, gene-gene interaction identification from genome-wide association studies, and prediction of regulatory elements from DNA and protein sequences.
- Sampling methods such as bagging and boosting are effective in dealing with data with small sample size, while random forests holds a unique advantage in dealing with data with high-dimensionality.
- Emerging ensemble methods such as ensemble of support vector machines, meta-ensemble, and ensemble of heterogeneous classification algorithms are promising directions for more accurate classification in bioinformatics.
- Ensemble based feature selection is a promising approach for feature selection and biomarker identification in bioinformatics.

Acknowledgement

We thank Professor Joachim Gudmundsson for critical comments and constructive suggestions which have greatly improve the early version of this article. We also thank Georgina Wilcox for editing the article. Pengyi Yang is supported by the NICTA International Postgraduate Award (NIPA) and the NICTA Research Project Award (NRPA).

References

[1] Larrañaga P, Calvo B, Santana R, Bielza C, Galdiano J, Inza I, et al. Machine learning in bioinformatics. Briefings in Bioinformatics. 2003; 7(1):86–112.

[2] Allison DB, Cui X, Page GP, Sabripour M. Microarray data analysis: from disarray to consolidation and consensus. Nature Reviews Genetics. 2006; 7(1):55–66.

[3] Aebersold R, Mann M. Mass spectrometry-based proteomics. Nature. 2003; 422(6928):198–207.

[4] Hirschhorn J, Daly M. Genome-wide association studies for common diseases and complex traits. Nature Reviews Genetics. 2005; 6(2):95–108.

[5] Zeng J, Zhu S, Yan H. Towards accurate human promoter recognition: a review of currently used sequence features and classification methods. Briefings in Bioinformatics. 2009; 10(5):498–508.

[6] Valencia A, Pazos F. Computational methods for the prediction of protein interactions. Current Opinion in Structural Biology. 2002; 12(3):368–373.

[7] Jones DT. Protein structure prediction in genomics. Briefings in Bioinformatics. 2001; 2(2):111–125.

[8] Dietterich TG. Ensemble methods in machine learning. In: Proceedings of Multiple Classifier System. vol. 1857. Springer; 2000. pp. 1–15.

[9] Dietterich TG. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting and Randomization. Machine Learning. 2000; 40:139–158.

[10] Breiman L. Bagging predictors. Machine Learning. 1996; 26(2):123–140.

[11] Freund Y, Schapire R. Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth National Conference on Machine Learning; 1996. pp. 148–156.

[12] Breiman L. Random Forests. Machine Learning. 2001; 45(1):5–32.

[13] Kuncheva L. Combining Pattern Classifiers: Methods and Algorithms. Wiley; 2004.

[14] Webb GI, Zheng Z. Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques. IEEE Transactions on Knowledge and Data Engineering. 2004; 16(8):980–991.

[15] Breiman L. Arcing classifiers (with discussion). The Annals of Statistics. 1998; 26(3):801–849.

[16] Schapire RE, Freund Y, Bartlett P, Lee WS. Boosting the margin: A new explanation for the effectiveness of voting methods. The Annals of Statistics. 1998; 26(5):1651–1686.

[17] Tsymbal A, Pechenizkiy M, Cunningham P. Diversity in search strategies for ensemble feature selection. Information Fusion. 2005; 6:83–98.

[18] Wolpert DH. Stacked generalization. Neural Networks. 1992; 5(2):241–259.

[19] Kuncheva LI. Switching between selection and fusion in combining classifiers: An experiment. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. 2002; 32(2):146–156.

[20] Gama J, Brazdil P. Cascade generalization. Machine Learning. 2000; 41(3):315–343.

[21] Lam L, Suen Y. Application of majority voting to pattern recognition: an analysis of its behaviour and performance. IEEE Transactions on Systems, Man, and Cybernetics. 1997; 27:553–568.

[22] Bahler D, Navarro L. Methods for Combining Heterogeneous Sets of Classifiers. In: Proceedings of the 17th National Conference on Artificial Intelligence (AAAI), Workshop on New Research Problems for Machine Learning; 2000. .

[23] Kang H, Kim K, J K. A framework for probabilistic combination of multiple classifiers at an abstract level. Engineering Applications of Artificial Intelligence. 1997; 10(4):379–385.

[24] Kittler J, Hatef M, Duin RP, Mates J. On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1998; 20:226–239.

[25] Asyali MH, Colak D, Demirkaya O, Inan MS. Gene expression profile classification: a review. Current Bioinformatics. 2006; 1(1):55–73.

[26] Somorjai RL, Dolenko B, Baumgartner R, Crow JE, Moore JH. Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. Bioinformatics. 2003; 19:1484–1491.

[27] Saeys Y, Lnza I, Larrañaga P. A review of feature selection techniques in bioinformatics. Bioinformatics. 2007; 23(19):2507–2517.

[28] Hilario M, Kalousis A. Approaches to dimensionality reduction in proteomic biomarker studies. Briefings in Bioinformatics. 2008; 9(2):102–118.

[29] Braga-Neto U, Dougherty E. Is cross-validation valid for small-sample microarray classification? Bioinformatics. 2004; 20(3):374–380.

[30] Ben-Dor A, Bruhn L, Friedman N, Nachman I, Schummer M, Yakhini Z. Tissue Classification with Gene Expression Profiles. Journal of Computational Biology. 2000; 7(3-4):559–583.

[31] Dudoit S, Fridlyand J, Speed T. Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association. 2002; 97:77–87.

[32] Dettling M, Bühlmann P. Boosting for tumor classification with gene expression data. Bioinformatics. 2003; 19(9):1061–1069.

[33] Long P. Boosting and Microarray Data. Machine Learning. 2003; 53:31–44.

[34] Tan A, Gilbert D. Ensemble machine learning on gene expression data for cancer classification. Applied Bioinformatics. 2003; 2(3 Suppl):S75–S83.

[35] Qu Y, Adam B, Yasui Y, Ward M, Cazares L, Schellhammer P, et al. Boosted Decision Tree Analysis of Surface-enhanced Laser Desorption/Ionization Mass Spectral Serum Profiles Discriminates Prostate Cancer from Noncancer Patients. Clinical Chemistry. 2002; 48(10):1835–1843.

[36] Wu B, Abbott T, Fishman D, McMurray W, Mor G, Stone K, et al. Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. Bioinformatics. 2003; 19(13):1636–1643.

[37] Gertheiss J, Tutz G. Supervised feature selection in mass spectrometry-based proteomic profiling by blockwise boosting. Bioinformatics. 2009; 25(8):1076–1077.

[38] Yasui Y, Pepe M, Thompson M, Adam B, Wright GJ, Qu Y, et al. A data-analytic strategy for protein biomarker discovery: profiling of high-dimensional proteomic data for cancer detection. Biostatistics. 2003; 4(3):449–463.

[39] Lee J, Lee J, Park M, Song S. An extensive comparison of recent classification tools applied to microarray data. Computational Statistics & Data Analysis. 2005; 48:869–885.

[40] Díaz-Uriarte R, de Andrés S. Gene selection and classification of microarray data using random forest. BMC Bioinformatics. 2006; 7:3.

[41] Izmirlian G. Application of the random forest classification algorithm to a SELDI-TOF proteomics study in the setting of a cancer prevention trial. Annals of the New York Academy of Sciences. 2004; 1020:154–174.

[42] Zhang H, Yu C, Singer B. Cell and tumor classification using gene expression data: Construction of forests. Proceedings of National Academy Science. 2003; 100(7):4168–4172.

[43] Geurts P, Fillet M, Seny D, Meuwis M, Malaise M, Merville M, et al. Proteomic mass spectra classification using decision tree based ensemble methods. Bioinformatics. 2005; 21(15):3138–3145.

[44] Dougherty ER, Sima C, Hanczar B, Braga-Neto UM. Performance of Error Estimators for Classification. Current Bioinformatics. 2010; 5(1):53–67.

[45] Ge G, Wong G. Classification of premalignant pancreatic cancer mass-spectrometry data using decision tree ensembles. BMC Bioinformatics. 2008; 9:275.

[46] Statnikov A, Wang L, Aliferis C. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics. 2008; 9:319.

[47] Curtis RK, Orešič M, Vidal-Puig A. Pathways to the analysis of microarray data. TRENDS in Biotechnology. 2005; 23(8):429–435.

[48] Pang H, Lin A, Holford M, Enerson BE, Lu B, Lawton MP, et al. Pathway analysis using random forests classification and regression. Bioinformatics. 2006; 22(16):2028–2036.

[49] Kanehisa M, Araki M, Goto S, Hattori M, Hirakawa M, Itoh M, et al. KEGG for linking genomes to life and the environment. Nucleic Acids Research. 2008; 36(Database issue):D480–D484.

[50] Montana G. Statistical methods in genetics. Briefings in Bioinformatics. 2006; 7(3):297–308.

[51] Cordell JH. Detecting gene-gene interactions that underlie human diseases. Nature Reviews Genetics. 2009; 10:392–404.

[52] Zhang H, Bonney G. Use of classification trees for association studies. Genetic Epidemiology. 2000; 19(4):323–332.

[53] Huang J, Lin A, Narasimhan B, Quertermous T, Hsiung CA, Ho LT, et al. Tree-structured supervised learning and the genetics of hypertension. Proceedings of the National Academy of Sciences. 2004; 101(29):10529–10534.

[54] Cook N, Zee R, Ridker P. Tree and spline based association analysis of gene-gene interaction models for ischemic stroke. Statistics in Medicine. 2004; 23(9):1439–1453.

[55] Ye Y, Zhong X, Zhang H. A genome-wide tree-and forest-based association analysis of comorbidity of alcoholism and smoking. BMC Genetics. 2005; 6(Suppl 1):S135.

[56] Zhang Z, Zhang S, Wong MY, Wareham NJ, Sha Q. An ensemble learning approach jointly modeling main and interaction effects in genetic association studies. Genetic Epidemiology. 2008; 32(4):285–300.

[57] McKinney BA, Reif DM, Ritchie MD, Moore JH. Machine learning for detecting gene-gene interactions: a review. Applied Bioinformatics. 2006; 5(2):77–88.

[58] Bureau A, Dupuis J, Hayward B, Falls K, Van Eerdewegh P. Mapping complex traits using Random Forests. BMC genetics. 2003; 4(Suppl 1):S64.

[59] Bureau A, Dupuis J, Falls K, Lunetta K, Hayward B, Keith T, et al. Identifying SNPs predictive of phenotype using random forests. Genetic Epidemiology. 2005; 28(2):171–182.

[60] Lunetta KL, Hayward LB, Segal J, Van Eerdewegh P. Screening large-scale association study data: exploiting interactions using random forests. BMC Genetics. 2004; 5(1):32.

[61] Chen X, Liu C, Zhang M, Zhang H. A forest-based approach to identifying gene and gene-gene interactions. PNAS. 2007; 104:19199–19203.

[62] Klein RJ, Zeiss C, Chew EY, Tsai JY, Sackler RS, Haynes C, et al. Complement factor H polymorphism in age-related macular degeneration. Science. 2005; 308(5720):385.

[63] Meng Y, Yu Y, Cupples L, Farrer L, Lunetta K. Performance of random forest when SNPs are in linkage disequilibrium. BMC Bioinformatics. 2009; 10:78.

[64] Jiang R, Tang W, Wu X, Fu W. A random forest approach to the detection of epistatic interactions in case-control studies. BMC Bioinformatics. 2009; 10(Suppl 1):S65.

[65] Zhang M. Computational analyses of eukaryotic promoters. BMC Bioinformatics. 2007; 8(Suppl 6):S3.

[66] Down TA, Hubbard TJP. Computational detection and location of transcription start sites in mammalian genomic DNA. Genome Research. 2002; 12(3):458–461.

[67] Blon N, Ponten T, Gupta R, Brunak S. Prediction of post-translational glycosilation and phosphorylation of proteins from aminoacid sequence. PROTEOMICS-Clinical Applications. 2004; 4(6):1633–1649.

[68] Hong P, Liu XS, Zhou Q, Lu X, Liu JS, Wong WH. A boosting approach for motif modeling using ChIP-chip data. Bioinformatics. 2005; 21(11):2636–2643.

[69] Xie X, Wu S, Lam KM, Yan H. PromoterExplorer: an effective promoter identification method based on the AdaBoost algorithm. Bioinformatics. 2006; 22(22):2722–2728.

[70] Zhao X, Xuan Z, Zhang M. Boosting with stumps for predicting transcription start sites. Genome Biology. 2007; 8(2):R17.

[71] Wang X, Xuan Z, Zhao X, Li Y, Zhang MQ. High-resolution human core-promoter prediction with CoreBoost_HM. Genome Research. 2009; 19(2):266–275.

[72] Gordon JJ, Towsey MW, Hogan JM, Mathews SA, Timms P. Improved prediction of bacterial transcription start sites. Bioinformatics. 2006; 22(2):142.

[73] Hamby SE, Hirst JD. Prediction of glycosylation sites using random forests. BMC Bioinformatics. 2008; 9(1):500.

[74] Caragea C, Sinapov J, Silvescu A, Dobbs D, Honavar V. Glycosylation site prediction using ensembles of Support Vector Machine classifiers. BMC Bioinformatics. 2007; 8(1):438.

[75] Yoo PD, Ho YS, Zhou BB, Zomaya AY. SiteSeek: Post-translational modification analysis using adaptive locality-effective kernel methods and new profiles. BMC Bioinformatics. 2008; 9(1):272.

[76] Guan Y, Myers C, Hess D, Barutcuoglu Z, Caudy A, Troyanskaya O. Predicting gene function in a hierarchical context with an ensemble of classifiers. Genome Biology. 2008; 9:S3.

[77] Shen HB, Chou KC. Ensemble classifier for protein fold pattern recognition. Bioinformatics. 2006; 22(14):1717–1722.

[78] Melvin I, Weston J, Leslie CS, Noble WS. Combining classifiers for improved classification of proteins from sequence or structure. BMC Bioinformatics. 2008; 9(1):389.

[79] Lee B, Shin M, Oh Y, Oh H, Ryu K. Identification of protein functions using a machine-learning approach based on sequence-derived properties. Proteome Science. 2009; 7:27.

[80] Wang SQ, Yang J, Chou KC. Using stacked generalization to predict membrane protein types based on pseudo-amino acid composition. Journal of Theoretical Biology. 2006; 242(4):941–946.

[81] Chen XW, Liu M. Prediction of protein-protein interactions using random decision forest framework. Bioinformatics. 2005; 21(24):4394–4400.

[82] Deng L, Guan J, Dong Q, Zhou S. Prediction of protein-protein interaction sites using an ensemble method. BMC Bioinformatics. 2009; 10(1):426.

[83] Wu CC, Asgharzadeh S, Triche TJ, D'Argenio DZ. Prediction of human functional genetic networks from heterogeneous data using RVM-based ensemble learning. Bioinformatics. 2010; 26(6):807.

[84] Altay G, Emmert-Streib F. Revealing differences in gene network inference algorithms on the network level by ensemble methods. Bioinformatics. 2010; 26(14):1738.

[85] Caragea C, Sinapov J, Silvescu A, Dobbs D. Glycosylation site prediction using ensemble of support vector machine classifiers. BMC Bioinformatics. 2007; 8:438.

[86] Peng Y. A novel ensemble machine learning for robust microarray data classification. Computers in Biology and Medicine. 2006; 36:553–573.

[87] Dettling M. BagBoosting for tumor classification with gene expression data. Bioinformatics. 2004; 20(18):3583–3593.

[88] Liu KH, Xu CG. A genetic programming-based approach to the classification of multiclass microarray datasets. Bioinformatics. 2009; 25(3):331–337.

[89] Bhanot G, Alexe G, Venkataraghavan B, Levine AJ. A robust meta-classification strategy for cancer detection from MS data. Proteomics. 2006; 6(2):592–604.

[90] Kedarisetti KD, Kurgan L, Dick S. Classifier ensembles for protein structural class prediction with varying homology. Biochemical and Biophysical Research Communications. 2006; 348(3):981–988.

[91] Hassan MR, Hossain MM, Bailey J, Macintyre G, Ho J, Ramamohanarao K. A voting approach to identify a small number of highly predictive genes using multiple classifiers. BMC Bioinformatics. 2009; 10(Suppl 1):S19.

[92] Yang P, Zhou B, Zhang Z, Zomaya A. A multi-filter enhanced genetic ensemble system for gene selection and sample classification of microarray data. BMC Bioinformatics. 2010; 11(Suppl 1):S5.

[93] Yang P, Ho JWK, Zomaya AY, Zhou BB. A genetic ensemble approach for gene-gene interaction identification. BMC Bioinformatics. 2010; 11(1):524.

[94] Liu B, Cui Q, Jiang T, Ma S. A combinational feature selection and ensemble neural network method for classification of gene expression data. BMC Bioinformatics. 2004; 5:136.

[95] Koziol JA, Feng AC, Jia Z, Wang Y, Goodison S, McClelland M, et al. The wisdom of the commons: ensemble tree classifiers for prostate cancer prognosis. Bioinformatics. 2009; 25(1):54–60.

[96] Amaratunga D, Cabrera J, Lee Y. Enriched random forests. Bioinformatics. 2008; 24(18):2010–2014.

[97] Yanover C, Singh M, Zaslavsky E. M are better than one: an ensemble-based motif finder and its application to regulatory element prediction. Bioinformatics. 2009; 25(7):868–874.

[98] Yanover C, Weiss Y. Finding the AI Most Probable Configurations Using Loopy Belief Propagation. In: Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference. The MIT Press; 2004. p. 289.

[99] Armañanzas R, Inza I, Larrañaga P. Detecting reliable gene interactions by a hierarchy of Bayesian network classifiers. Computer Methods and Programs in Biomedicine. 2008; 91(2):110–121.

[100] Robles V, Larranaga P, Pena JM, Menasalvas E, Pérez MS, Herves V, et al. Bayesian network multiclassifiers for protein secondary structure prediction. Artificial Intelligence in Medicine. 2004; 31(2):117–136.

[101] Hu J, Yang YD, Kihara D. EMD: an ensemble algorithm for discovering regulatory motifs in DNA sequences. BMC Bioinformatics. 2006; 7(1):342.

[102] Wijaya E, Yiu SM, Son NT, Kanagasabai R, Sung WK. MotifVoter: a novel ensemble method for fine-grained integration of generic motif finders. Bioinformatics. 2008; 24(20):2288–2295.

[103] Boulesteix AL, Slawski M. Stability and aggregation of ranked gene lists. Briefings in Bioinformatics. 2009; 10(5):556–568.

[104] Jiang H, Deng Y, Chen HS, Tao L, Sha Q, Chen J, et al. Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. BMC Bioinformatics. 2004; 5(1):81.

[105] Levner I. Feature selection and nearest centroid classification for protein mass spectrometry. BMC Bioinformatics. 2005; 6(1):68.

[106] Saeys Y, Abeel T, Van de Peer Y. Robust feature selection using ensemble feature selection techniques. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases. Part II. vol. 5212. Springer; 2008. pp. 313–325.

[107] Dutkowski J, Gambin A. On consensus biomarker selection. BMC Bioinformatics. 2007; 8(Suppl 5):S5.

[108] Zhang Z, Yang P, Wu X, Zhang C. An agent-based hybrid system for microarray data analysis. IEEE Intelligent Systems. 2009; 24(5):53–63.

[109] Abeel T, Helleputte T, Van de Peer Y, Dupont P, Saeys Y. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. Bioinformatics. 2010; 26(3):392–398.

[110] Netzer M, Millonig G, Osl M, Pfeifer B, Praun S, Villinger J, et al. A new ensemble-based algorithm for identifying breath gas marker candidates in liver disease using ion molecule reaction mass spectrometry. Bioinformatics. 2009; 25(7):941–947.

[111] Yang YH, Xiao Y, Segal MR. Identifying differentially expressed genes from microarray experiments via statistic synthesis. Bioinformatics. 2005; 21(7):1084–1093.

[112] Chan D, Bridges SM, Burgess SC. In: An Ensemble Method for Identifying Robust Features for Biomarker Discovery. Chapman & Hall; 2007. pp. 377–392.

[113] Blum AL, Langley P. Selection of relevant features and examples in machine learning. Artificial Intelligence. 1997; 97(1-2):245–271.

[114] Guyon I, Elisseeff A. An introduction to variable and feature selection. The Journal of Machine Learning Research. 2003; 3:1157–1182.

[115] Dash M, Liu H. Feature selection for classification. Intelligent Data Analysis. 1997; 1(3):131–156.

[116] Saeys Y, Inza I, Larrañaga P. A review of feature selection techniques in bioinformatics. Bioinformatics. 2007; 23(19):2507–2517.

[117] Xing EP, Jordan MI, Karp RM. Feature selection for high-dimensional genomic microarray data. In: Proceedings of the Eighteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc.; 2001. pp. 601–608.

[118] Zhang K, Qin ZS, Liu JS, Chen T, Waterman MS, Sun F. Haplotype block partitioning and tag SNP selection using genotype data and their applications to association studies. Genome Research. 2004; 14(5):908.

[119] Kalousis A, Prados J, Hilario M. Stability of feature selection algorithms: a study on high-dimensional spaces. Knowledge and Information Systems. 2007; 12(1):95–116.

[120] Jurman G, Merler S, Barla A, Paoli S, Galea A, Furlanello C. Algebraic stability indicators for ranked lists in molecular profiling. Bioinformatics. 2008; 24(2):258.

[121] Smyth G. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Statistical Applications in Genetics and Molecular Biology. 2004; 3:Article3.

[122] Moore J, White B. Tuning ReliefF for genome-wide genetic analysis. Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics. 2007; pp. 166–175.

[123] Kohavi R, John GH. Wrappers for feature subset selection. Artificial Intelligence. 1997; 97(1-2):273–324.

[124] Li L, Weinberg CR, Darden TA, Pedersen LG. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics. 2001; 17(12):1131–1142.

[125] Li L, Umbach DM, Terry P, Taylor JA. Application of the GA/KNN method to SELDI proteomics data. Bioinformatics. 2004; 20(10):1638.

[126] Jirapech-Umpai T, Aitken S. Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. BMC Bioinformatics. 2005; 6(1):148.

[127] Ruiz R, Riquelme JC, Aguilar-Ruiz JS. Incremental wrapper-based gene selection from microarray data for cancer classification. Pattern Recognition. 2006; 39(12):2383–2392.

[128] Potamias G, Koumakis L, Moustakis V. Gene selection via discretized gene-expression profiles and greedy feature-elimination. Methods and Applications of Artificial Intelligence. 2004; pp. 256–266.

[129] Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. Machine Learning. 2002; 46(1):389–422.

[130] Díaz-Uriarte R, De Andres SA. Gene selection and classification of microarray data using random forest. BMC Bioinformatics. 2006; 7(1):3.

[131] Dunne K, Cunningham P, Azuaje F. 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; .

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

[133] Alon U, Barkai N, Notterman D, Gish K, Ybarra S, Mack D, et al. 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. 1999; pp. 6745–6750.

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

[135] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter. 2009; 11(1):10–18.

[136] He Z, Yu W. Stable feature selection for biomarker discovery. Computational Biology and Chemistry. 2010; 34(4):215–225.

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

[138] Jong K, Mary J, Cornuéjols A, Marchiori E, Sebag M. Ensemble feature ranking. In: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases. Springer-Verlag New York, Inc.; 2004. pp. 267–278.

[139] Dietterich TG. Ensemble methods in machine learning. In: Proceedings of the First International Workshop on Multiple Classifier Systems. Springer-Verlag; 2000. pp. 1–15.

[140] Tsymbal A, Pechenizkiy M, Cunningham P. Diversity in search strategies for ensemble feature selection. Information Fusion. 2005; 6(1):83–98.

[141] DeConde R, Hawley S, Falcon S, Clegg N, Knudsen B, Etzioni R. Combining results of microarray experiments: a rank aggregation approach. Statistical Applications in Genetics and Molecular Biology. 2006; 5:Article15.

[142] Kalousis A, Prados J, Hilario M. Stability of feature selection algorithms. In: Proceedings of the Fifth IEEE International Conference on Data Mining. IEEE; 2005. pp. 218–225.

[143] Jeffery IB, Higgins DG, Culhane AC. Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data. BMC Bioinformatics. 2006; 7(1):359.

[144] Kuncheva LI. A stability index for feature selection. In: Proceedings of the 25th IASTED International Multi-Conference: Artificial Intelligence and Applications. ACTA Press; 2007. pp. 390–395.