Health Technol. (2017) 7:141–152
DOI 10.1007/s12553-016-0157-z

ORIGINAL PAPER

Splice site identification in human genome using random forest

Elham Pashaei[1] · Mustafa Ozen[2] · Nizamettin Aydin[1]

Received: 28 June 2016 / Accepted: 25 November 2016 / Published online: 2 December 2016

Abstract Gene identification has been an increasingly important task due to developments of Human Genome Project. Splice site prediction lies at the heart of identifying human genes, thus development of new methods which detect the splice site accurately is crucial. Machine learning classifiers are utilized to detect the splice sites. Performance of those classifiers mainly depends on DNA encoding methods (feature extraction) and feature selection. The feature extraction methods try to capture as much information as the DNA sequences have, while the feature selection methods provide useful biological knowledge by cleaning out the redundant information. According to the literature, Markovian models are popular encoding methods and the support vector machine (SVM) is known as the best algorithm for classification of splice sites. However, random forest (RF) may outperform the SVM in this domain using those Markovian encoding methods. In this study, performance of RF has been investigated as feature selection and classification in splice site domain. We proposed three methods, namely MM1-RF, MM2-RF and MCM-RF by combining RF with first order Markov Model (MM1), second order Markov model (MM2), and Markov Chain Model (MCM). We compared the performance of the RF with the SVM competitively on HS3D and NN269 benchmark datasets. Also, we evaluated the efficiency of the proposed methods with other current state of arts methods such as Reduced MM1-SVM, SVM-B and LVMM2. The experimental results show that the RF outperforms the SVM when the same Markovian encoding methods are used on both donor and acceptor datasets. Furthermore, the RF classifier performs much faster than the SVM classifier in detecting the splice sites.

Keywords Splice site prediction · DNA encoding methods · Random Forest classifier · Gene detection

1 Introduction

Biological sequence data has been increasing rapidly during the past few decades, so there is a crucial need of effective methods to detect genes [1, 2]. Despite of many efforts, the issue has been not solved satisfactorily yet [3]. Accurate splice site identification is essential in gene detection. In eukaryotic genomes, each gene is composed of exons and introns. During DNA transcription only exons of the gene, which contain codes for proteins are transcribed into mRNAs. [4]. The term splice site refers to boundary between exon and intron [5]. While the intron-exon junction with consensus dinucleotide AG is called acceptor splice site, donor splice site refers to exon-intron junction with consensus dinucleotide GT (see Fig. 1) [3]. In DNA sequence, splice site prediction is a search problem for finding donor and acceptor boundaries.

To predict the splice site, approximately all of the proposed methods consist of three main steps; proper encoding schema

This article is part of the Topical collection on *Systems Medicine*

✉ Nizamettin Aydin
naydin@yildiz.edu.tr

Elham Pashaei
elham.pashaei@std.yildiz.edu.tr

Mustafa Ozen
mozen@bcm.edu

[1] Department of Computer Engineering, Yildiz Technical University, Istanbul, Turkey

[2] Department of Pathology & Immunology, Baylor College of Medicine, Houston 77030, TX, USA

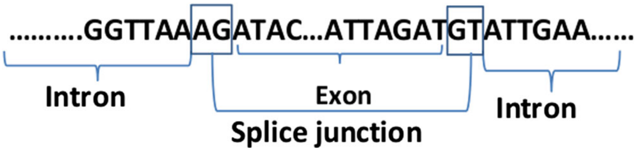

Fig. 1 Schematic representation of the splice junction site [5]

(feature extraction), feature selection (optionally), and classification. Machine learning methods are used to detect splice site (classification step). The input of machine learning classifiers is numerical, whereas the information of DNA sequences is given as strings. Therefore, encoding the DNA sequence into numbers is initial and main task of splice site prediction (feature extraction step) [6]. The probabilistic encoding approaches such as the zero order Markov model (MM0), the first order Markov model (MM1), the second order Markov model (MM2), and the Markov Chain Model (MCM) are so famous and high usage methods [7–16].

In biology, where structures are described by a large number of features as splice sites, the feature selection is an important step towards the classification task. It provides useful biological knowledge and allows for a faster and better classification. Feature selection techniques by considering the method's output can be divided into two groups; wrapper methods and filter methods [17, 18]. The wrapper methods pick up the feature subset based on classifiers performance. However, the filter methods assess the relevance of features via univariate statistical criteria instead of cross-validation performance. So, the wrapper methods give better performance result than filter methods due to taking into account features dependencies and directly interacting with the classifier. However, they are computationally more expensive than filter approaches [18]. On the other hand, the filter methods are known as the fast, rapidly scalable and efficient feature selection approaches in bioinformatics [17, 18]. There are two types of filter methods, univariate and multivariate methods. Most filter methods in the literature are univariate [17]. Multivariate filter methods can find relationships among the features, whereas univariate methods consider each feature individually. Therefore, multivariate filter methods can not disclose mutual information between features [19]. There are many various wrapper and filter approaches in the literature. Particle swarm optimization (PSO), genetic algorithms (GA), sequential forward and backward selection are some examples of the wrapper approach, while chi-square, correlation coefficient, Fisher score (F-score) feature ranking are some examples of filter approaches. There are few specific works where feature selection techniques have been used in splice site prediction domain. Principle feature selection (PFA) is a multivariate filter method that has been employed by Maji [15] in Human splice site prediction. F-score feature ranking [9, 14] and Estimated distribution algorithm (EDA) ranking methods [20] are two univariate filter methods that have been applied on human and plants splice sites, respectively. Also the EDA has been utilized as a wrapper approach in [21] which has shown good performance in plant splice site prediction.

Random forests (RF) are among the most popular machine learning methods due to their relatively good performance. They also provide method for feature selection [22–24]. The random forest feature ranking (variable importance) has been used in various domain such as integrated analysis of multiple data type [25], biomarker discovery [26] and multi-label classification [27]. In this study we investigate the ability of random forest feature ranking methods on the splice site prediction domain.

Various successful computational methods such as support vector machine [1, 6, 8, 9, 14, 28], decision trees [29, 30], hidden Markov model [13, 31], artificial neural network [2, 32–34] and Bayesian network [35, 36] have been developed to recognize splice junction of DNA sequences. Among them, SVM is the most popular classifier method [5]. Baten [8] used MM1 encoding method to extract the features of splice sites sequences and give them to SVM as the input for classifying splice sites. Reduced MM1-SVM [9] was developed using F-score feature ranking method to choose a subset of more informative MM1 parameters for SVM to predict splice site. Zhang [6] constructed a mapping method from Bayes' rule and integrated it with linear SVM (SVM-B) to predicted splice sites. A length-variable Markov model (LVMM) [13] is developed by employing the MM2 encoding. The method can choose a particular subset of features to predict a candidate splice site according to the ratio of likelihood at each position. Despite of the high accuracy that LVMM method produces, determining the method's threshold parameters is not easy task [14]. In [15], a hybrid approach using second order Markov model and SVM with principle feature analysis (MM2F-SVM) as a new feature selection method has been proposed. The MCM encoding method [12] that is combination of MM1 and MM2 has provided inputs for SVM classifier in [16]. Despite the presence of these methods, splice site identification remains still a major bottleneck in gene detection domain due to existing complex dependencies between the bases around splice site [37]. Therefore, development of accurate methods to identify splice site junction continue [2].

This study is concerned with RF for feature selection [23] and classification in splice site prediction domain. The performance of RF ranking method has been compared with F-score feature ranking [38] by using the learning curve concept. Liu [39] and Kocev [27] have remarked on the use of learning curves to show the effect of adding features when a list of ordered features is provided. We have investigated their effect on HS3D datasets with the goal of using a small number of features to achieve better classification performance.

Due to its high performance, SVM classifier is frequently used in prediction of splice sites. However, some parameters of SVM classifier such as penalty parameter, the kernel type, and kernel parameters, must be tuned. Parameter tuning can be time-consuming when there are multiple parameters involved in the training. So, one should be cautious whether SVM is a suitable method to genome-wide splice sites prediction or not [13]. In this study, we have combined RF as an efficient and fast classifier with three predefined encoding methods (MM1 [8], MM2 [15], MCM [3, 12, 16]) and compared their results with the SVM. We have also investigated effect of our methods on H3SD and NN269 datasets and have evaluated efficiency of proposed methods by making a comparison with some current methods such as MM1-SVM [9], SVM-B [6], LVMM [13], MCM-SVM [16], and MM2F-SVM [15].

The remainder of the paper is organized as follows. In Section 2, Materials and methods are described. Experimental results are explained in Section 3. Section 4 provides the conclusion.

2 Materials and methods

2.1 Splice sites datasets

Experiments have been performed on the *Homo sapiens* Splice Site Data set (HS3D) [40], which is composed of 2796 confirmed true donor sites, 2880 confirmed true acceptor sites, 271,937 false donor sites, and 329,374 false acceptor sites. The performance of proposed methods are examined on both donor and acceptor sites separately. Each splice site sequence consists of 140 nucleotides with the consensus nucleotides AG at position 69 and 70 and consensus nucleotide GT at position 71 and 72 for acceptor sites and donor sites, respectively. Balanced (1:1) and unbalanced (1:10) datasets have been formed by selecting all the true splice sites for both of them. The ratio between number of true splice site and randomly selected false splice site in the balanced dataset is the same, whereas in unbalanced dataset number of randomly selected false splice sites is 10 times more than true splice sites.

We have performed an extra evaluation on the NN269 dataset [10] to estimate the reproducibility and consistency of our method. The dataset has been gathered from 269 human genes that are composed of 1324 true acceptor sites, 5552 false acceptor sites, 1324 true donor sites, and 4922 false donor sites. The NN269 dataset has been divided into two subsets: the acceptor dataset and the donor dataset. The training dataset for acceptor (donor) site are made up of 1116 true acceptor (donor) sites and 4672 false acceptor (4140 false donor) sites. The test dataset contains 208 true acceptor (donor) sites and 881 false acceptor (782 false donor) sites. We evaluate the efficiency of the proposed methods on acceptor sites and donor sites separately. The length of the sequences in acceptor splice site is 90 nucleotides whereas donor splice sites have the length of 15 nucleotides. The consensus dinucleotide AG in acceptor splice site is at positions 69 and 70 and the consensus nucleotides GT in donor splice site is at positions 8 and 9.

2.2 Markovian based encoding methods

To do classification analysis on splice sites, DNA sequences should be represented as feature vectors. Different encoding methods are applied to DNA sequences to extract associated features. Each encoding method tries to provide as much information as sequences have. The performance of a classifier used in splice sites prediction highly depends on the DNA encoding methods. So, effective DNA encoding methods for extracting feature vectors from DNA sequences are essential. In this study, MM1 encoding [8], MM2 encoding [15], and MCM [12, 16] encoding have been used. The Markov model describes a sequence of possible states, in which the probability of each state depends only on the preceding states.

Consider a sequence $(s_1, s_2, \dots, s_n)$ of length n. The nucleotide s_i is a realization of the i th state variable in Markov chain. Each state is characterized by a position-specific probability parameter. The set of parameters in first order Markov model and second order Markov model are $\{P(s_i| s_{i-1})\}$ and $\{P(s_i| s_{i-1}, s_{i-2})\}$, respectively. The estimation of the model parameters is calculated by (1)

$$P(s_i|s_{i-1}, \dots, s_{i-k}) = \frac{N(s_{i-k}, \dots, s_i)}{N(s_{i-k}, \dots, s_{i-1})} \quad (1)$$

where k denotes the order of Markov model and $N(s_{i-k}, \dots, s_i)$ shows the occurrence number of $(s_{i-k}, \dots, s_i)$. In this study $k = 1$ and $k = 2$ have been chosen for MM1 and MM2. As it is mentioned in [8], to create Markov model only true splice site sequences are considered.

The MCM was earlier used by Lio in [12] and again was employed recently in [3, 16]. This encoding method utilizes both MM1 and MM2 encoding methods. Each sequence is broken down into three parts: signal segment (S^S), upstream segment (S^U), and downstream segment (S^D), as shown in Fig. 2. The signal segment is encoded by MM1 and the model is denoted by M_S. The upstream segments and downstream segments are encoded using MM2 and denoted by M_U and M_D, respectively. We also define a false model M_F to

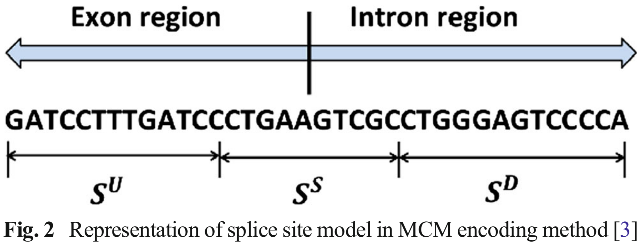

Fig. 2 Representation of splice site model in MCM encoding method [3]

characterize the signal segment for false splice sites. The final model is combination of them, that is (M_U, M_S, M_F , M_D). We have set l_U=30, l_S=47, and l_D= 63 bp for donor sites, l_U= 48, l_S=21, and l_D=69 bp for acceptor site in the HS3D dataset, while we have adjusted l_U=3, l_S=9, and l_D= 3 bp for donor sites and l_U= 52, l_S=19, and l_D=19 bp for acceptor site in the NN269 dataset.

2.3 Random forest classifiers

The RF, which has been introduced by Breiman in 2001 [41], is an ensemble classification algorithm based on decision trees. Each tree in the forest is trained by randomly selecting samples with replacement (bootstrap) from total samples of the original dataset. The rest of the samples are used as the test set. A single decision tree uses randomly *m* number of features from total *M* features in splitting each node (*mtry*). A random forest with *k* decision tree (*ntree*) repeats above procedure for each decision tree and final classification is obtained by voting result of these *k* decision trees on testing data. Figure 3 describes the steps of Random Forest algorithms. We have implemented Random Forest algorithm using "Random Forest" package in R software. The Random Forest has two parameters for tuning namely "*mtry*" and "*ntree*". They are number of features to choose at each node for splitting and number of trees to be grown in the forest respectively. In this study, "*mtry*" is equal to √*M*, while "*ntree*" is equal to 500 (default value) on the HS3D dataset. The value of "*ntree*" has been set to 530 for the NN269 dataset.

2.4 Support vector machine classifier

SVM [42] is the most important learning machine that has been used in many domains due to its excellent classification accuracy. The SVM aims to find a maximal margin hyperplane to separate classes. The kernel function are used to map data to a higher dimensional space for learning non-linearly separable functions. New instances are classified according to the direction of the hyperplane they belong to [43]. The accuracy of the SVM largely depends on the proper chosen kernel and its parameters. This study has adopted radial basis function (RBF) kernel and utilized SVM of "e1071" package, which is an interface of LIBSVM in R. We have used grid-based search method to find optimal parameters (C- penalty parameter and γ-gamma).

2.5 Fisher score feature ranking method

The feature ranking methods typically assign a weight to each feature and rank them accordingly. Then informative features can be selected and low-scoring features are removed. F-score is a simple univariate filter approach, which is used for ranking features according to their discriminative powers. Given training instance x_i, $i = 1 , \ldots , l$, the F-score of the *j*th attribute is calculated by:

$$F(j) = \frac{\left(\overline{x}_j^{(+)} - \overline{x}_j\right)^2 + \left(\overline{x}_j^{(-)} - \overline{x}_j\right)^2}{\text{variance}\left(\overline{x}_j^{(+)}\right) + \text{variance}\left(\overline{x}_j^{(-)}\right)} \tag{2}$$

where $\overline{x}_j^{(+)}$, $\overline{x}_j^{(-)}$ and $\overline{x}_j$ are the average of the *j*th attribute of the positive, negative and whole datasets, respectively. The numerator indicates the inter class variance, while the sum of the variance inside each class is shown by the denominator. High F-score value of an attribute demonstrates that this attribute has more discriminative power [38].

2.6 Random Forest feature ranking method

Ranking of variables can be obtained by utilizing the mechanism of random forest. Each tree in the random forest is constructed on 2/3 of the training data which are drawn randomly with replacement (bootstrap). The split in each node of the trees is selected from subset of variables (features). After building trees of forest, each tree is tested on the 1/3 of the samples which have not been selected for bootstrap. These samples are called the Out-Of-Bag (OOB) instances and error of predictive performance of them is shown with *Err*(*OOB*). The OOB is used for ranking variables by permuting each variable (*j*) one-by-one in OOB dataset of all the trees and

Fig. 3 Algorithm of random forest classifier

```
Input: training dataset $D_{N*M}$ , number of trees ($k$) in forest, size of feature subset ($m$) that is considered at each node during tree construction
Begin
 For $i = 1$ to $k$ do
    •  Draw a bootstrap sample of size $N$ from the training dataset.
    •  Grow a random-forest tree $T_i$ to the 2/3 of bootstrapped data, by recursively repeating the following steps for each terminal node of the tree until the minimum node size $n_{min}$ is reached.
          ▪  Select $m$ features at random from total $M$ features.
          ▪  Pick the best feature/split-point among the $m$.
          ▪  Split the node into two daughter nodes.
 End For
Output the ensemble of trees $\{T_i\}_1^k$.
End
To make a prediction at the new point $x$:
 $\hat{C}_{RF}^k(x) = majority\ vote\ \{\hat{C}_i(x)\}_1^k$ , let $\hat{C}_i(x)$ be the class prediction of the $i$th tree in RF.
```

calculating error of predictive performance of the permuted version of OOB data (Err_j)). Subtraction of these errors is calculated at the next step. Ultimately, the average error of subtraction results and associated variances are measured. Figure 4 explains algorithm of calculating ranking of feature using RF clearly. The "*FSelector*" R package has been used for implementation of RF feature ranking method. More detailed explanation on RF can be found in [27, 44]

2.7 Classification performance evaluation metrics

In this study, sensitivity (S_n), specificity (S_p), a global accuracy (Q^9), Matthew's correlation coefficients (Mcc), area under ROC curve (AUC), and F-measure have been used as the performance measure. These measures are defined as follows:

$$S_n = TP/(TP + FN) \tag{3}$$

$$S_p = TN/(TN + FP) \tag{4}$$

$$Q^9 = \frac{(1+q^9)}{2}/2$$
$$q^9 = \begin{cases} \dfrac{(TN-FP)}{(TN+FP)} & \text{if} \quad (TP+FN)=0 \\ \dfrac{(TP-FN)}{(TP+FN)} & \text{if} \quad (TN+FP)=0 \\ 1-\sqrt{2\left[\left(\dfrac{FN}{TP+FN}\right)^2+\left(\dfrac{FP}{TN+FP}\right)^2\right]} & \text{if } (TP+FN)\neq 0 \\ & \text{and } (TN+FP)\neq 0 \end{cases} \tag{5}$$

$$Mcc = \frac{(TP{*}TN)-(FP{*}FN)}{\sqrt{(TP{*}FN){*}(TN{*}FP){*}(TP{*}FP){*}(TN{*}FN)}} \tag{6}$$

$$F\text{−measure} = 2{*}\frac{TP/(TP+FP){*}S_n}{TP/(TP+FP)+S_n} \tag{7}$$

where *TP*, *FP*, *TN* and *FN* show the number of true positives, false positives, true negatives and false negatives, respectively. Larger values of the S_n, S_p, Q^9, Mcc, and $F-$*measure* indicate better classification performance.

The Receiver Operator Characteristic (ROC) curve are obtained by plotting sensitivity against 1-specificity and is used for visualizing the performance of the binary classifier. The area under ROC curve (AUC) is utilized for summarizing the performance in a single number. On the other hand, plotting True Positive Rate versus the False Positive Rate gives precision recall curve (PRC) and the area under PRC curve (auPRC) has again summarized the performance in a single number. The increment in the value of AUC and auPRC lead to a more accurate model performance.

2.8 Cross-validation design

The10-fold cross-validation has been used to evaluate the performance of our methods on the HS3D dataset [13, 14]. For this, we have divided the data sets into 10 equal size parts (folds). After the dataset has been separated into parts, a model is made using 9 of the folds as a training set and the remaining fold as a test set. This process is replicated 10 times with a different test set each time. Furthermore, we have repeated each experiment 5 times to increase the reliability of the evaluation. Each time, different folds are generated randomly and average of 5 independent repeats has been reported.

Due to existence of the large difference between number of true and false sites in unbalanced (1:10) datasets of HS3D, the performance of the classifiers tends to be biased towards the majority class [45]. To overcome this problem, undersampling technique [46, 47] has been used. For this purpose, we only modified the training set (9 folds out of 10) by considering that each fold contains the same proportion of number of true sites versus number of false sites in unbalance dataset.

Fig. 4 Algorithm for feature ranking via random forest

```
Input: training dataset $D_{N*M}$ , number of trees ($k$) in forest, size of feature subset ($m$) that is considered at each
node during tree construction
Output: Importance of each feature
Begin
For $i = 1$ to $k$ do
  • Draw a bootstrap sample of size $N$ from the training dataset.
  • Grow a random-forest tree $T_i$ to the $2/3$ of bootstrapped data
  • Give the leftover $1/3$ of samples (called OOB) to the tree $T_i$ , and calculate the error rate $Err(OBB)$
      For $j = 1$ to $M$ do // for each feature $j \in M$
          •  Permute the value of feature $j$ randomly for the OOB samples
          •  Compute the error rate for permuted version of OBB samples $Err_j$ using tree $T_i$
          •  Calculate $d_j = Err_j - Err(OBB)$
      End For
End For
For $j = 1$ to $M$ do
  • Aggregate total error rate from all trees and calculate variance for each feature
    $\hat{d} = \frac{1}{k}\sum_{i=1}^{k} d_i^j$ and $S_d = \frac{1}{k-1}\sum_{i=1}^{k}(d_i^j - \hat{d})^2$
  • Calculate variable importance  $v_{j=} \hat{d}/S_d$
End For
End
```

The training is performed on all the true sites by randomly selecting the same number of false sites on training set without modifying the test set.

For NN269 dataset, in order to tune parameters of SVM, we divided training dataset into 10 equally sized data fold. Each fold contains the same proportion of true versus false sequences. For each parameter combination, we used 9 out of 10 folds and evaluated the methods on the remaining fold. We selected the model with the highest average of auPRC on 10 evaluation sets. Then this best model was trained on the complete training dataset. The ultimate evaluation was performed on the corresponding independent test sets. According to [48], when the binary classifier on imbalanced dataset is evaluated, the auPRC is more informative than AUC. So, we focused on auPRC measure for model selection of SVM.

2.9 Statistical comparison among classifiers

It is important to determine whether the differences between results of classifiers are statistically significant or not when they are compared. Therefore we utilized t-test to assess significance of differences in classification performance. The null hypothesis of the test is that there is no difference between performance of the SVM and the RF. A significance level $\alpha = 0.01$ has been used in this study.

2.10 Proposed methods to assess performance of RF

RF as feature ranking The proposed procedure consists of two steps (see Fig. 5) for investigating RF feature ranking approach in Human splice site detection. At the first step, we have applied RF feature ranking method to train dataset. Consequently, a value is assigned to each feature indicating importance of each feature in classification accuracy. Then, we sort them according to their values decreasingly. At the second step, we evaluate the ranking by performing a stepwise feature subset evaluation, which is used to provide the *learning curve.* For this purpose, we select the top-*k* ranked features from the ordered variables. Then, we evaluate performance of the classifier on chosen subset feature and constructed *forward feature addition curve* (FFA).

RF as classifier Splice site is subdivided into two separate classification problems: acceptor splice site classification and donor splice site classification. We try to identify whether a candidate splice site is true splice site (positive) or not (negative) for both classification problems. So, two different models are constructed for them to make prediction. These models consist of two phases: feature extraction using encoding scheme and classification. The proposed methods MM1-RF, MM2-RF, and MCM-RF utilize Markovian encoding approaches MM1, MM2, and MCM to provide features and use RF for classification. The steps of models are outlined in Fig. 6.

3 Results

3.1 Efficiency of RF as feature ranking approach

Performance of selected attributes on balanced and unbalanced datasets have been shown in Fig. 7. From the figure, it is possible to state that the accuracy of simple MM1-SVM has been improved by using feature ranking approaches.

By considering balanced datasets (see Fig. 7a and b), it can be seen that both feature ranking methods have approximately the same accuracy on their optimal points. Additionally the optimal points of both are equal in balanced acceptor and donor sites. The optimal point of balanced acceptor dataset and balanced donor dataset have been achieved by choosing 60% and 30% of top features using both of the feature ranking methods, respectively. Considering results for unbalanced datasets shown in the second row of the Fig. 7, result of the RF ranking in acceptor sites (see Fig. 7c) is higher than the F-Score and optimal point has been obtained using fewer numbers of attributes. In unbalanced donor splice sites (See Fig. 7d) F-Score shows better performance than the RF ranking method. So, on 4 datasets, the RF ranking method shows two equal, one win and one failure on its performance. As a result, on average it can be concluded that the RF feature ranking method is a good candidate for performing feature selection as preprocessing part on splice sites prediction methods.

Fig. 5 Algorithm of providing forward feature addition curve using random forest

```
Input: The provided training data $D_{N*M}$, number of total features $M$
Output: Forward feature addition curve (FFA)
Begin
 Compute the RF score of importance for all the feature. $R = \{I_1, I_2, \ldots, I_M\}$ is the vector of obtained feature ranking.
 Order the features in decreasing order of importance
 For $i = 1$ to 10 do
    • select $k$-top ranked feature from $R$ and accordingly carry out feature selection on training set, $k = (i * 10 * M)/100$
    • Apply SVM on the training set $D_{N*k}$ to learn the prediction model
    • Use the model to make prediction on the test set with the chosen $k$ features(calculate $Q^9$)
    • Return $Q^9$ measurement for drawing FFA curve
 End For
End
```

Fig. 6 Algorithm of the proposed splice site prediction methods MM1-RF, MM2-RF and MCM-RF

```
Input: The candidate splice site sequences, $\{S_1, S_2, \dots, S_N\}$
Output: Labels of unknown sequences
Begin
  For $i = 1$ to $N$ do
      • Model $S_i$ using one of the proposed Markovian encoding methods (MM1, MM2 or MCM). The
        Output is a vector of features, $F_i = (f_1, f_2, f_3, \dots)$
  End For
  Apply RF on the training set of the extracted features $\{F_1, F_2, \dots, F_N\}$ to learn the prediction model
  Use the model to make prediction on the test sequences of splice sites
End
```

3.2 Efficiency of RF as classifier

The performance results of classification have been shown in Table 1. Since different training data are obtained due to employing different encoding methods, we considered each row of the table as an independent dataset. Therefore, our experiment utilized 18 different datasets (9 for the acceptor sites, 9 for the donor sites). The performance was estimated using various measures. However, we preferred $F-measure$ to make statistical comparison (reported *P*-value) between SVM and RF. We should take into account that we could not carry out statistical evaluation on NN269 dataset due to default separation between training set and test set. However, we consider their results as significant when the difference in F-measure became more than 1.50% between SVM and RF.

According to the results, the RF outperforms the SVM significantly in 8 datasets and nominally in 4 datasets. However, the SVM outperforms the RF significantly in 4 datasets and

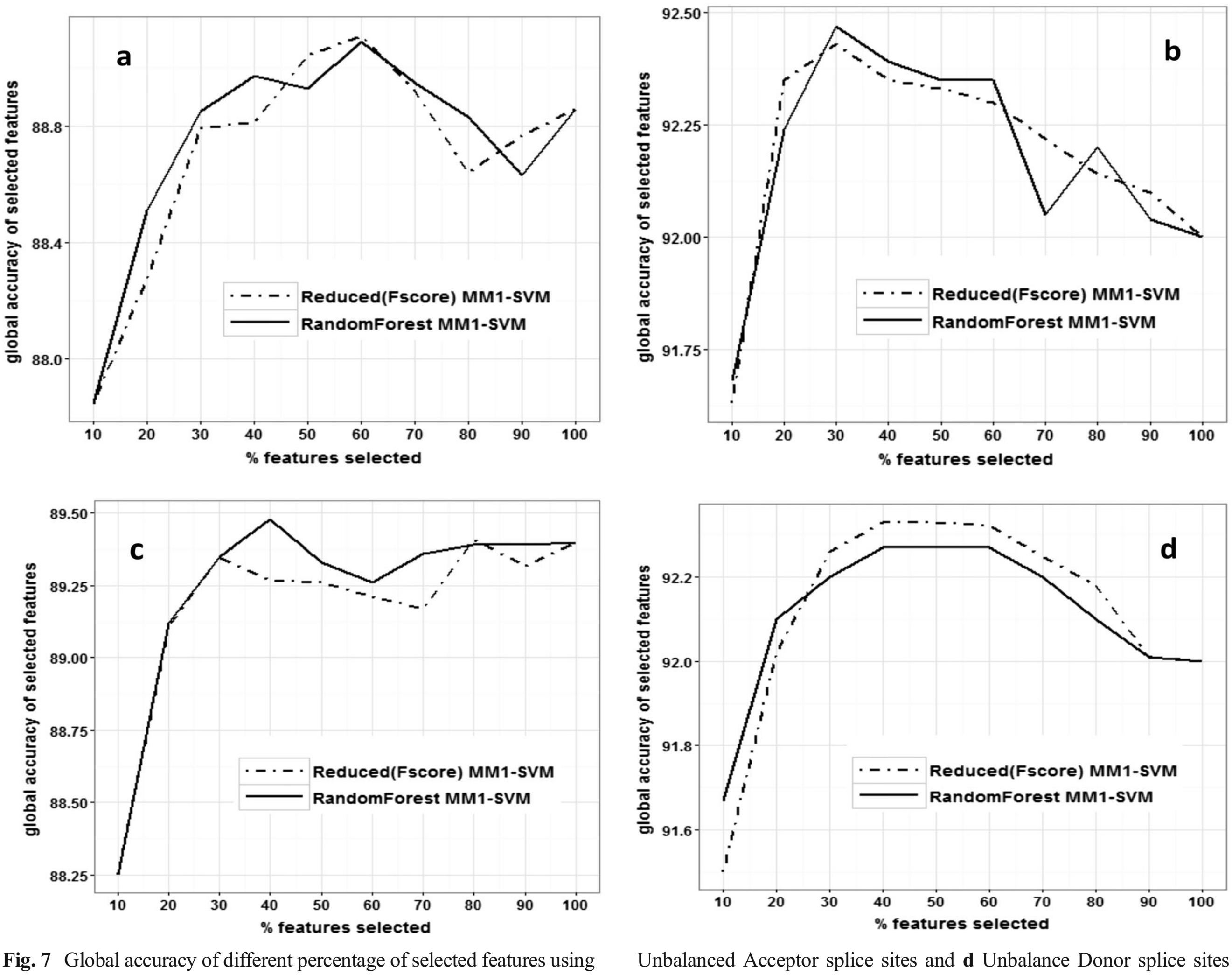

Fig. 7 Global accuracy of different percentage of selected features using F-score feature ranking and random forest feature ranking methods on **a** Balanced Acceptor splice sites, **b** Balanced Donor splice sites, **c** Unbalanced Acceptor splice sites and **d** Unbalance Donor splice sites datasets for assessing performance of MM1-SVM method

Table 1 Comparison of classification performance of SVMs and RFs using Markovian encoding methods

Dataset	Encoding methods	Support Vector Machine (SVM)							Random Forest (RF)							Nominally superior method	P- Value
		S_n	***S_p***	***Q^9***	***Mcc***	AUC	F-measure	CPU time(s)	***S_n***	***S_p***	***Q^9***	***Mcc***	AUC	F-measure	CPU time (s)		
Balanced Acceptor (HS3D)	MM1	90.22	87.79	88.86	78.06	95.43	89.14	1556.25	91.69	89.10	90.24	80.84	96.36	90.52	484.89	RF	**0.00095**
	MM2	91.42	88.17	89.60	79.65	96.00	89.95	1274.84	92.16	89.47	90.65	81.68	96.62	90.94	500.98	RF	**0.00824**
	MCM	92.87	88.35	90.28	81.32	96.52	90.82	1301.68	91.92	89.67	90.66	81.63	96.62	90.90	557.02	RF	0.6504
Balanced Donor (HS3D)	MM1	93.27	91.12	92.01	84.43	97.15	92.27	1064.89	95.40	91.00	92.78	86.50	97.72	93.35	472.25	RF	0.0187
	MM2	93.37	91.59	92.36	85.00	97.34	92.55	793.94	95.77	91.29	93.09	87.17	97.97	93.67	467.84	RF	**0.00812**
	MCM	95.67	93.16	94.20	88.94	98.18	94.52	1006.01	95.44	92.00	93.41	87.52	97.93	93.84	622.96	SVM	0.0605
Unbalanced Acceptor (HS3D)	MM1	90.77	88.27	89.39	58.04	95.78	58.95	1678.18	91.87	89.52	90.59	61.09	96.52	61.96	528.20	RF	**0.00006**
	MM2	91.90	88.97	90.29	60.04	96.25	60.83	1332.50	92.17	89.99	90.97	62.21	96.73	63.09	525.89	RF	**0.00008**
	MCM	92.88	89.00	90.69	60.70	96.66	61.34	1388.43	91.94	89.69	90.71	61.48	96.64	62.56	582.91	RF	**0.00994**
Unbalanced Donor (HS3D)	MM1	93.24	91.00	91.99	65.02	97.18	65.54	1146.47	95.06	90.72	92.52	65.55	97.75	66.07	472.68	RF	0.8506
	MM2	93.41	91.37	92.27	65.94	97.38	66.80	837.84	95.70	91.55	93.25	67.79	97.96	68.33	454.23	RF	0.0265
	MCM	95.36	93.14	94.09	71.49	98.22	72.30	1059.76	94.93	92.12	93.32	68.65	98.02	69.38	620.38	SVM	**0.00004**
Unbalanced Acceptor (NN269)	MM1	75.96	96.71	82.84	75.74	97.52	80.00	336.60	70.19	97.50	78.85	73.67	97.33	77.66	28.91	SVM	-
	MM2	79.33	97.05	85.23	78.94	97.92	82.71	301.08	71.63	97.96	79.89	75.92	98.01	79.47	28.08	SVM	-
	MCM	80.77	97.73	86.31	81.65	98.43	84.85	297.36	69.71	98.30	78.55	75.51	97.69	78.80	34.82	SVM	-
Unbalanced Donor (NN269)	MM1	83.65	96.16	88.13	80.39	97.82	84.47	54.95	85.58	96.55	89.51	82.56	97.86	86.20	4.37	RF	-
	MM2	83.17	96.16	87.80	80.05	97.33	84.18	52.21	85.10	96.68	89.20	82.51	97.84	86.13	4.20	RF	-
	MCM	90.87	95.65	92.85	84.37	98.06	87.70	58.24	88.01	96.42	91.21	83.96	97.70	87.35	5.18	SVM	-

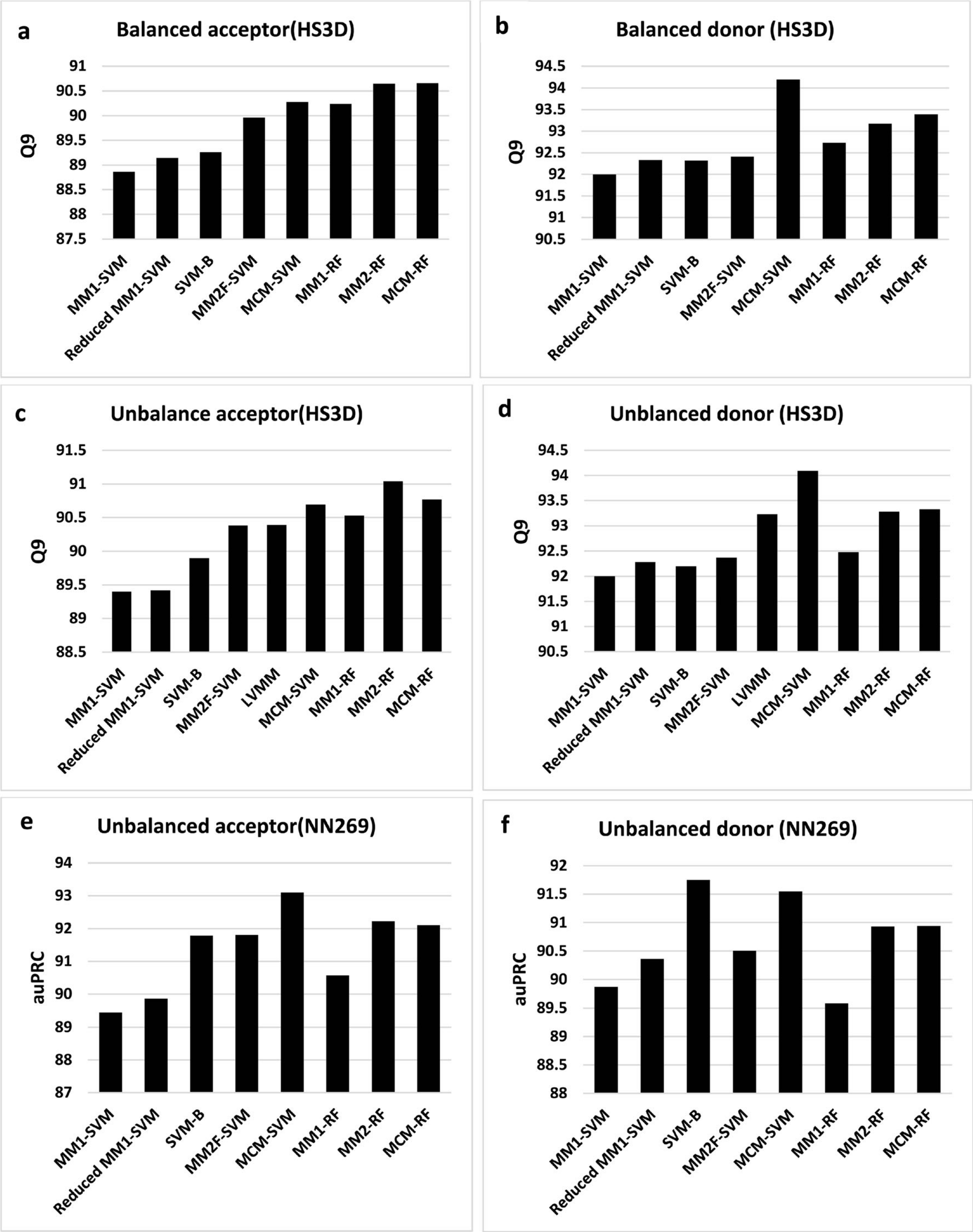

Fig. 8 Classification performance of the different state of art methods for both HS3D and NN269 datasets

nominally in 2 datasets. So, considering 18 datasets, overall RF performs better than SVM in 12 datasets. In terms of computational efficiency, as can be seen from CPU time column in the Table 1, the RF performed much faster than the SVM due to parameter tuning process involved in the SVM.

In addition, the classification results of proposed methods MM1-RF, MM2-RF and MCM-RF compared with these of MM1-SVM [8], Reduced MM1-SVM [9], SVM-B [6], LVMM2 [13], MM2F-SVM [15] and MCM-SVM [16] methods using Q^9 criteria for HS3D dataset and *auPRC* for NN269 dataset in Fig. 8. The result of the LVMM2 was taken from [13].

From Fig. 8, considering both balanced datasets, the proposed method MM1-RF outperformed MM1-SVM, Reduced MM1-

SVM, SVM-B and MM2F-SVM for both acceptor (Fig. 8a) and donor splice site (Fig. 8b), but could not show better performance than MCM-SVM. Two other proposed methods, MM2-RF and MCM-RF performed better than MM1-RF for both acceptor and donor sites. In balanced acceptor splice site (Fig. 8a), MM2-RF and MCM-RF showed the same performance and both of them could outperform other methods. In balanced donor site (Fig. 8b), MCM-RF performed better than MM2-RF and MM1-RF and could outperform all of the other methods except MCM-SVM. Considering unbalanced acceptor dataset (Fig. 8c), we can see that MM1-RF outperformed the MM1-SVM, Reduced MM1-SVM and SVM-B and produce comparable result with LVMM and MM2F-SVM. The MCM-RF method performed better than MM1-RF and could outperform LVMM and MM2F-SVM. The MM2-RF method performed better than MCM-RF and outperformed all methods significantly and stood out as the best method on unbalanced acceptor splice sites. In the unbalance donor site (Fig. 8d), the MM1-RF outperformed MM1-SVM, Reduced MM1-SVM, SVM-B and MM2F-SVM. The MM2-RF performed better than MM1-RF and could produce comparable results with LVMM. The MCM-RF performed slightly better than the MM2-RF and could outperform all the methods except the MCM-SVM same as the MM2-RF. In comparison to LVMM2, the proposed methods MM2-RF and MCM-RF performed slightly better than LVMM2. However, determining the associated threshold parameters of the LVMM [13] are difficult [14]. The proposed method has less complexity in comparison to LVMM2. The overall performance comparison of the proposed methods can be summarized in this way. Considering the balanced acceptor dataset, MM2-RF and MCM-RF showed the best performance. The MCM-SVM method illustrated better accuracy than the proposed methods on balanced donor splice sites. Considering unbalanced datasets, the MM2-RF outperformed all the methods on acceptor site and again MCM-SVM showed higher accuracy in unbalanced donor sites. We can state that our proposed methods are definitely more suitable for acceptor sites than donor sites. Additionally, considering performance of RF along with SVM using the same encoding methods, the proposed methods in most of the cases performed better.

In order to estimate the consistency of the proposed methods, we performed an additional evaluation on the NN269 dataset. For acceptor sites (Fig. 8e), auPRC of the MM1-RF is better than MM1-SVM and Reduced MM1-SVM. Besides, the MM2-RF performed better than MM2F-SVM and SVM-B. The MCM-RF outperformed all of the methods but MCM-SVM performed better than the proposed methods. For the donor sites (Fig. 8f), the auPRC of MM1-RF method is lower than other available models. The MM2-RF and MCM-RF showed the same accuracy in term of auPRC. Both of them outperformed all methods except SVM-B and MCM-SVM methods. Overall, the proposed methods produced good results for NN269 dataset.

4 Conclusion and discussion

In this study, we study RF as a new classifier and feature selection method in Human splice site prediction domain. Since a large number of features are used to describe structures or processes in biology, the elimination of irrelevant and redundant information provide useful biological knowledge for human experts. F-score feature ranking method is a simple and efficient method that is used in splice site prediction domain frequently. We have investigated efficiency of RF feature ranking method by comparing it with F-score to show capability of RF as a feature selection in Human splice sites identification. The results show that RF feature ranking is useful method in human splice sites prediction.

SVM has been most commonly used in prediction of splice sites due to its high performance. But existing of the parameters that have to be set before using it, such as penalty parameter, the kernel type and kernel parameters make it time-consuming process, causing to question whether SVM is a suitable method to genome-wide splice sites prediction [13]. In this study we employ RF as another extremely successful classifier. One of main advantages of RF-based methods in comparison to SVM-based methods is that it does not need tuning step in contrary to SVM and it is really fast with high performance.

By combining RF with three up-to-date encoding methods (MM1, MM2, and MCM), we show that the proposed methods perform approximately the same and often better than the SVM-based methods. In addition, the proposed methods are simple, fast, easy to use and can be applied to large scale Human Genome data for identifying splice sites. As a future study, these methods can also be utilized in identification of other regulatory regions such as translation initiation sites and promoters.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Sonnenburg S, Schweikert G, Philips P, Behr J, Ratsch G. Accurate splice site prediction using support vector machines. BMC Bioinformatics. 2007;88(Suppl 10):S7.
2. Bin W, Jing Z. A novel artificial neural network and an improved particle swarm optimization used in splice site prediction. J Appl Comput Mathematics. 2014;3(4) doi:10.4172/2168-9679.1000166.
3. Nassa T, Singh S, Goel N. Splice site detection in DNA sequences using probabilistic neural network. Intern J Comp Appl(IJCA). 2013;76(4):1–4.

4. Salekdeh AY, Wiese KC. Improving splice-junctions classification employing a novel encoding schema and decision-tree. Evol Comput (CEC). 2011:1302–7. doi:10.1109/CEC.2011.5949766.
5. Bari AG, Reaz MR, Choi HJ, Jeong BS. Survey on nucleotide encoding techniques and SVM kernel Design for Human Splice Site Prediction. Interdisciplinary Bio Central. 2012;4(14):1–6. doi:10.4051/ibc.2012.4.4.0014.
6. Zhang Y, Chu C-H, Chen Y, Zha H, Ji X. Splice site prediction using support vector machines with a Bayes kernel. Expert Syst Appl. 2006;30(1):73–81.
7. Burge C, Karlin S. Predictions of complete gene structures in human genomic DNA. J Mol Biol. 1997;9(5):499–509.
8. Baten A, Chang B, Halgamuge S, Li J. Splice site identification using probabilistic parameters and SVM classification. BMC Bioinformatics. 2006;7(Suppl 5):S15.
9. Baten A, Halgamuge S, Chang B. Fast splice site detection using information content and feature reduction. BMC Bioinformatics. 2008;9(Suppl 12):S8.
10. Reese M, Eeckman F, Kupl D, Haussler D. Improved splice site detection in genie. J Comput Biol. 1997;4(3): 311–24.
11. Hebsgaard SM, Korning PG, Tolstrup N, Engelbrecht J, Rouzé P, Brunak S. Splice site prediction in *Arabidopsis thaliana* pre-mRNA by combining local and global sequence information. Nucleic Acids Res. 1996;24:3439–52.
12. Loi HS, Rajapakse JC. Splice site detection with a higher-order Markov model implemented on a neural network. Genome Informatics. 2003;14:64–72.
13. Zhang Q, Peng Q, Zhang Q, Yan Y, Li K, Li J. Splice site prediction of human genome using length-variable Markov model and feature selection. Expert Syst Appl. 2010;37(4):2771–82.
14. Wei D, Zhang H, Wei Y, Jiang Q. A novel splice site prediction method using support vector machine. J Comput Inf Syst. 2013;9(20):8053–60.
15. Maji S, Garg D. Hybrid approach using SVM and MM2 in splice site junction identification. Curr Bioinforma. 2014;9(1):76–85.
16. Goel N, Singh S, Aseri TC. An improved method for splice site prediction in DNA sequences using support vector machines. Procedia Comp Sci. 2015;57:358–67. doi:10.1016/j.procs.2015.07.350.
17. Ang JC, Mirzal A, Haron H, Hamed HNA. Supervised, unsupervised, and semi-supervised feature selection: a review on Gene selection. IEEE/ACM Transac Comp Biol Bioinformatics. 2016;13(5):971–89.
18. Kumari B, Swarnkar T. Filter versus wrapper feature subset selection in large dimensionality micro array: a review. Intern J Comp Sci Inform Technol (IJCSIT). 2011;2(3):1048–53.
19. Hira ZM, Gillies DF. A review of feature selection and feature extraction methods applied on microarray data. Adv Bioinforma. 2015;2015 doi:10.1155/2015/198363.
20. Saeys Y, Degroeve S, Aeyels D, Rouze P, Peer Y. Feature selection for splice site prediction: a new method using EDA-based feature ranking. BMC Bioinformatics. 2004;5(64) doi:10.1186/1471-2105-5-64.
21. Saeys Y, Degroeve S, Aeyels D, Van PD, Rouze P. Fast feature selection using a simple estimation of distribution algorithm: a case study on splice ste prediction. Bioinformatics. 2003;19(SUPPL2): 179–88.
22. Svetnik V, Liaw A, Tong C, editors. Variable Selection in Random Forest with Application to Quantitative Structure-Activity Relationship. Proceedings of the 7th Course on Ensemble Methods for Learning Machines. USA: Springer-Verlag; 2004.
23. Genuera R, Poggi JM, Malotc CT. Variable selection using random forests. Pattern Recognition Letters, Elsevier. 2010;31(14):2225–36.
24. Han L, Embrechts MJ, Szymanski B, Sternickel K, Ross A. Random Forests Feature Selection with Kernel Partial Least Squares: Detecting Ischemia from Magneto Cardiograms. Burges, Belgium: European Symposium on Artificial Neural Networks; 2006. p. 221–6.
25. Reif DM, Motsinger AA, McKinney BA, Crowe JE, Moore JH. Feature Selection using a Random Forests Classifier for the Integrated Analysis of Multiple Data Types. Symposium on Computational Intelligence and Bioinformatics and Computational Biology (CIBCB'06); Toronto: IEEE; 2006. p. 1–8. doi:10.1109/CIBCB.2006.330987.
26. Slavkov I, Zenko B, Dzeroski S. Evaluation method for feature rankings and their aggregations for biomarker discovery. In: JMLR Workshop and Conference Proceedings: Machine Learning in Systems Biology. 2010. vol. 8. p. 122–35.
27. Kocev D, Slavkov I, Dzeroski S, editors. Feature ranking for multi-label classication using predictive clustering trees. International Workshop on Solving Complex Machine Learning Problems with Ensemble Methods, in Conjunction with ECML/PKDD; 2013.
28. Wei D, Zhuang W, Jiang Q, Wei Y. A new classification method for human gene splice site prediction. In: He J, Liu X, Krupinski E, Xu G, editors. Health information science lecture notes in computer science. Heidelberg: Springer; 2012. p. 121–30.
29. Lopes HS, Lima CRE, Murata NJ. A configware approach for high-speed parallel analysis of genomic data. J Circuits Syst Comp. 2007;16:527–40.
30. Sun H, Peng Q, Zhang Q, Mou D. Splice site prediction based on characteristic of sequential motifs and C4.5 algorithm. In: 50th International Conference on Fuzzy Systems and Knowledge Discovery. Jinan Shandong: China IEEE; 2008. p. 417–22. doi:10.1109/FSKD.2008.331.
31. Yin M, Wang J. Effective hidden Markov models for detecting splicing junction sites in DNA sequences. Inf Sci. 2001;139:139–63.
32. Rajapakse J, Ho L. Markov encoding for detecting signals in genomic sequences. IEEE-ACM Transact Comp Biol Bioinform. 2005;2(2):131–42.
33. Marashi S, Goodarzi H, Sadeghi M, Eslahchi C, Pezeshk H. Importance of RNA secondary structure information for yeast donor and acceptor splice site prediction by neural networks. Comput Biol Chem. 2006;30(1):50–7.
34. Johansen O, Ryen T, Eftesol T, Kjosmoen T, Ruoff P. Splice site Predicton using artificial neural networks. In: Masulli F, Tagliaferri R, Verkhivker GM, editors. Computational intelligence methods for bioinformatics and biostatistics. Lecture notes in computer science. Heidelberg: Springer; 2009. p. 102–33.
35. Cai D, Delcher A, Kao B, Ksif S. Modeling splice sites with Bayes networks. Bioinformatics. 2000;16:152–8.
36. Chen T, Lu C, Li W. Prediction of splice sites with dependency graphs and their expanded bayesian networks. Bioinformatics. 2005;21:471–82.
37. Tsai K, Lin S, Shih S, Lai J, Chenn C. Genomic splice Sirte prediction algorithm based on nucleotide sequence pattern for RNA viruses. Comput Biol Chem. 2009;33:171–5.
38. Chen YW, Lin CJ. Combining SVMs with various feature selection strategies. In: Guyon I, Gunn S, Nikrevesh M, Zadeh L, editors. Feature extraction studies in fuzziness and soft computing. New York: Springer; 2006. p. 315–24.
39. Liu H, Motoda H. Feature selection for Knowlegde discovery and data mining. London: Kluwer Academic Publisher; 1998.

40. Pollastro P, Rampone S. HS3D, a dataset of homo sapies splice site regions, and its extraction procedure from a major public database. Inter J Modern Physics. 2002;C13(13):1105–17.
41. Breiman L. Random Forest. Mchine Learning. 2001;45(1):5–32. doi:10.1023/A:1010933404324.
42. Vapnik VN. Statistical learning theory. Adaptive and learning system for signal processing communications and control. New York: 1998.
43. Statnikov A, Wang L, Aliferis CF. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics. 2008;9:319.
44. Filimon A. Hedge fund fraud prediction using classication algorithms. Merlin: University of Zurich; 2011.
45. Lin WJ, Che JJ. Class-imbalanced classifiers for high-dimensional data. Brief Bioinform. 2012;14(1):13–26. doi:10.1093/bib/bbs006.
46. Ganganwar V. An overview of classification algorithms for imbalanced datasets. Intern J Emerg Technol Advance Eng(IJETAE). 2012;2(4):42–7.
47. Longadge R, Dongre SS, Malik L. Class imbalance problem in data mining: review. Intern J Comp Sci Net (IJCSN). 2013;2(1):83–7.
48. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS One. 2015;10(3) doi:10.1371/journal.pone.0118432.