JPPI Vol 9 No 1 (2019) 27 - 36

Jurnal Penelitian Pos dan Informatika
32a/E/KPT/2017

e-ISSN 2476-9266
p-ISSN: 2088-9402

Doi:10.17933/jppi.2019.090103

COMPARATION ANALYSIS OF ENSEMBLE TECHNIQUE WITH BOOSTING(XGBOOST) AND BAGGING(RANDOMFOREST) FOR CLASSIFY SPLICE JUNCTION DNA SEQUENCE CATEGORY

ANALISIS PEMBANDINGAN TEKNIK ENSEMBLE SECARA (XGBOOST) DAN BAGGING (RANDOMFOREST) PADA KLASIFIKASI KATEGORI SAMBATAN SEKUENS DNA

Iswaya Maalik S[1], Wisnu Ananta Kusuma[2], Sri Wahjuni[3]

[123] Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor
maalikiswaya@gmail.com

Naskah Diterima: 31 Oktober 2018; Direvisi : 4 Maret 2019; Disetujui : 5 Agustus 2019

Abstract

Bioinformatics research is currently undergoing a rapid growth, supported by the development of computation technology and algorithm. Ensemble decision tree is a common method for classifying large and complex dataset such as DNA sequence. Combining the implementation of two classification methods like xgboost and random Forest with ensemble technique might improve the accuracy result on classifying DNA Sequence splice junction type. With 96.24% accuracy for xgboost and 95.11% for Random Forest, the study suggests that both methods, using the right parameter setting, are highly effective tools for classifying DNA sequence dataset. Analyzing both methods with their characteristics will give an overview on how they work to meet the needs in DNA splicing.

Keywords: DNA splice site junction, ensemble technique, extreme gradient boosting, grid search hyperparamater optimization, random forest.

INTRODUCTION

Researches in the fields of genome and genetics are facilitated with the computational technology and machine learning algorithm. Machine Learning (ML) uses machine to learn and recognize patterns to be able to make classifications and even predictions. The high level of accuracy make it easy for researchers to evaluate an experiment immediately and precisely at an inexpensive cost. This technology has been widely implemented in many fields related to genetics and genomics because it is considered to be able to interpret enormous genome dataset and has been used to describe a wide variety of varieties from the part of the genomic sequence (Libbrecth, 2015).

Biogenetic data is also related to the process of protein formation. There is a stage in the process of protein synthesis where deoxyribonucleic acid (DNA) is copied into ribonucleic acid (RNA). The copy resulted in unnecessary information which are carried to the final product, thus the RNA form is considered immature. Such information must be removed in order to produce functional products. RNA splicing process is done to eliminate information that is not needed. Exons are sequences of nucleotides that remain in the mature RNA, whereas introns are sequences that are removed. The classification of data refers to 2 types of splicing categories, namely the acceptor and donor categories. The acceptor is the border between the intron gene and the exon gene while the donor is the DNA sequence containing a border between the exon gene and the intron gene.

In the last decade, the pattern recognition algorithm for splice site junction has continued to develop. Among them are the weight matrix method (WMM), weight array method (WAM), maximal dependence decomposition (MDD), hidden markov model (HMM), artificial neural network (ANN), and support vector machine (SVM) which have been widely applied and implemented in some software (ZX Sun, 2008).

One of the common methods used in ML is the decision tree (DT). DT is able to extract information from a dataset into knowledge that is intuitive and easy to understand (Barros et al., 2012). DT algorithms has advantages over other learning algorithms, for example its endurance towards noise, low computational cost to produce a model, and ability to handle excessive features (Rokach and Maimon, 2005). DT classifiers are also considered to be very useful, efficient and commonly used to deal with data mining classification problems (Farid et al, 2014).

One of DT weaknesses on availability of training data with weak predictive values can be overcome by the application of ensemble techniques. The ensemble method is a learning algorithm that is developed from several classification or predictive models. Lately, the computing application in biology has seen an increase use of ensemble learning method because of its unique advantages in handling small sample sizes, high dimensions, and complex data structures (Yang et al 2010). However, ideally the availability of data and variations are needed for better accuracy because the size of determinant attributes variation in the classification contributes to the accuracy value to form prediction models in an ensemble (Hamed and Can, 2017). Two methods commonly used in ensemble techniques are boosting and bagging.

The boosting method is in the form of repeated weighting of the predictor. The boosting method used

is gradient boosting (GB) in the form of boosting by gradient descent. GB was first introduced by Friedman et al . (2001), one of the improvised algorithms is (xgboost) by Chen and Guestrin (2016). This extreme gradient boosting algorithm is very popular and it often wins the ML competition held by Kaggle.

Ensemble concept with bagging is done by combining many prediction values into one prediction value. One of the advantages of Bagging is that it can reduce prediction errors generated by a single DT . Random Forest (RF) is one of the DT methods that employ the bagging concept. RF uses predictor candidates randomly on each tree for training process and votes will be made for the entire tree formed.

The two ensemble techniques will be implemented in DNA sequences derived from the UCI machine learning repository. Tuning parameters is carried out to improve the accuracy of ML. The results of the implementation of both methods are then analyzed in terms of their performance. It is expected that the results of this analysis can provide an idea of how these methods real implementation of working mechanisms could assist research in the field of DNA splicing.

METHODOLOGY

This study compares testing on the models that are built using each method. Models were built using a computer device with Intel quad core specifications with 8GB of memory with Microsoft Windows 10 operating system. The software used to build the model is R programming language using the library caret, dplyr, XG Boost and RandomForest packages. Datasets were managed using the Notepad plus editor.

This study is carried out in 3 main stages, namely pre-process, the implementation of ensemble techniques to form models with training process with default parameters of each method, and then the results and performance were compared with test data. Evaluation is carried out by repeating the training and testing process several times with various configurations of number of iterations or trees that are built. Optimization also performed with other parameters in addition to the number of iterations or a tree with grid search method in greedy matter to obtain the value with maximum accuracy. The last step is to analyze the process time and accuracy of each model built. In order to obtain more in-depth information about the work mechanism of the ML is carried out with literature studies of related journals and papers. Details of the mechanism of this study are illustrated in the following chart.

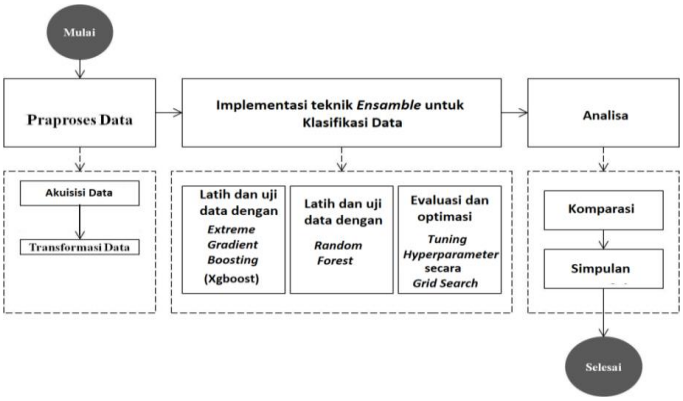


Figure 1. stages of research of the implementation of ensemble method on DNA sequence dataset

Data of this study is taken from Genbank 64.1 (ftp:://genbank.bio.net). The dataset "Primate splice-junction gene sequences (DNA) with associated imperfect domain theory" is a DNA sequence from primates in the form of splice-junction sequences (Lichman M., 2013). Data downloaded from the UCI machine learning is a nucleotide sequences labeled splice exon-intron category and the opposite intron-exon sequences and neither categories.

Data pre-process

The initial stage is to pre-process the data which includes data acquisition, coding in numerical values, conversion to matrix and distribution of training and test data. At the stage of data acquisition, the DNA sequence dataset compression file is downloaded via the internet at the address https://archive.ics.uci.edu/ml/machine-earningdatabases/molecular-biology/splice-junction-gene-sequences/splice.data.Z.

Table 1. Dataset description

Dataset characteristics	Number of attributes	Number of classes	Number of features	*Missing Value*
Sequential	*61*	3	3,190	none

Data extracted and converted into CSV format. Furthermore the data is divided into training and test data. Training data was 75% of the overall data of 2,392 record data training divided by the number of categories proportionally. The remaining 798 or 25% is used as test data.

Variables in DNA sequence consisting of a group categories of intron-exon (IE), Neither (N) and exon-introns (EI) while the nucleotide sequence is adenine (A), cytosine (C), guanine (G), and thymine (T). The DNA sequence code and categories were then categorized into a number value because XGBoost requires data in numerical form. There are no special requirements in coding, the important thing is that the values in the nucleotide code feature and label are unique. Information codification in shown in Table 2.

The EI category value is converted to 0, the N category is to 2 and last, the IE category is to 1. The values of the nucleotide adenine, cytosine, guanine, and thymine which are clearly defined are converted to 3, 4, 5 and 6. In a nucleotide sequence, not all types of base can be clearly defined, but the nucleotide have characters that characterize the value of the possible the nucleotide type. For nucleotides which have a possible value of coded "D" adenine, guanine, and thymine are converted to number 7. The type of nucleotide that has a probability of being adapted to four base types of N values is converted

to number 8. Nucleotides which may be cytosine or coded guanine "S" is converted to a value of 9. Whereas nucleotides which may be in the form of coded "R" denine or guanine are converted to number 0. There was only a little percentage of base types that are not clearly identified so that classification process was not affected. After making sure the dataset has been converted into a number value and the missing value is not found, then the data needs to be converted into a matrix.

Table 2. Codification to number

Code	information	conversion
EI	*Ekson – Intron*	0
IE	*Intron – Ekson*	1
N	*(Neither)*	2
A	*Adenin*	3
C	*Cytosine*	4
G	*Guanine*	5
T	*Thymine*	6
D	A atau G atau T	7
N	A atau G atau C atau T	2
S	C atau G	8
R	A atau G	9

Data classification using the ensemble method which is a learning algorithm built from several models of classification or predictor. The most commonly used ensemble techniques are boosting and bagging.

Bagging or bootstrap aggregating is an ML method built in an ensemble for stability and good accuracy in classification and regression. To prevent overfitting, the number of variants are reduced and it usually done in the form of decision tree with the application of the average value of generated model. The concept is to make the data sample D sizes n, and then produce new training data as many as m where each set of size n based on random data D with replacement of content data. Classifications are made based on these m samples. Each sample has a probability of $(1-1/n)^n$ to be selected as test data.

Random forest is a classification algorithm developed from the classification and regression tree (CART) method. This method optimizes the estimation process by bagging. Random forest is formed from many Decision Trees from sample data which have undergone training process. Before tree formation, the random feature selection stage is carried out. The results of the entire tree will be evaluated through voting. The basic concept of random forest is the implementation of bootstrap aggregating (bagging) method.

Boosting is an ensemble method which moves sequentially. The method is employed by combining weak predictor models to produce better predictive accuracy. For each iteration, models are resulted from the previous weighting process. Boosting focuses on new learning process on data with a low accuracy value produced in previous process and is carried out with a sequential training process. Incorrect data from the previous prediction is classified as "difficult" data and will be used for the next prediction process so that the accuracy value reaches a maximum point. After the whole prediction process is carried out, all models are merged. Boosting transforms a weak predictor model into a reliable complex predictor. The stages of this learning process are predicting for regression, calculation of errors of the residue, and learning process to process the residue.

One of the forms of ensemble implementations by boosting is gradient boosting

(GB). GB is a regression and classification algorithm that applies the ensemble concept of weak predictors and generally uses decision trees. Optimization process is carried out through boosting by optimizing the value of loss function. Gradient boosting combines weak predictors iteratively by minimizing the mean square error of the model where *error* $(\hat{y} - y)$ of model F and $\hat{y} = f(x)$. From each of the iteration process, a collection of hypotheses are produced, forming model and producing predictive value.

For illustration, Figure 2 shows the mechanism of a

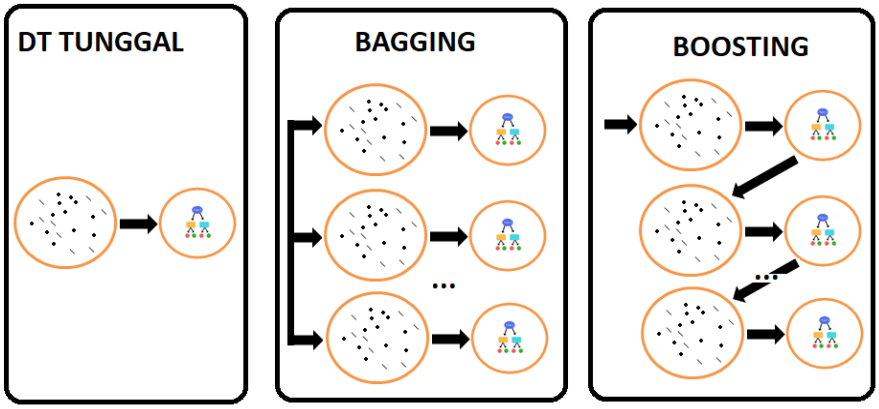


Figure 2. *Ensemble* on *decision tree*

single DT development that can be built by ensemble method, bagging and booting, in an optimization effort to obtain better accuracy value.

RESULTS AND DISCUSSION

Training process is carried out in the range of the number of trees, between 30 and 130. The number was obtained from the initial testing by measuring the error level of logloss and Mean Square Error (MSE) at a certain point whose graph is relatively stable.

The processing time is directly proportional to the number of trees. The more trees to be grown, the longer the time needed to carry out the classification process. For xgboost method, longer time is needed for processing than in the random forest. It happened because the xgboost mechanism operates sequentially while the random forest in parallel.

Accuracy level analysis for XGboost and Random Forest test process

After training process was conducted on training data, approximately 100 models of xgboost and random forest were produced, each of which has different parameters of numbers of tree or nround. Then, the next stage is testing all models built with the prepared test data during the data pre-process stage.

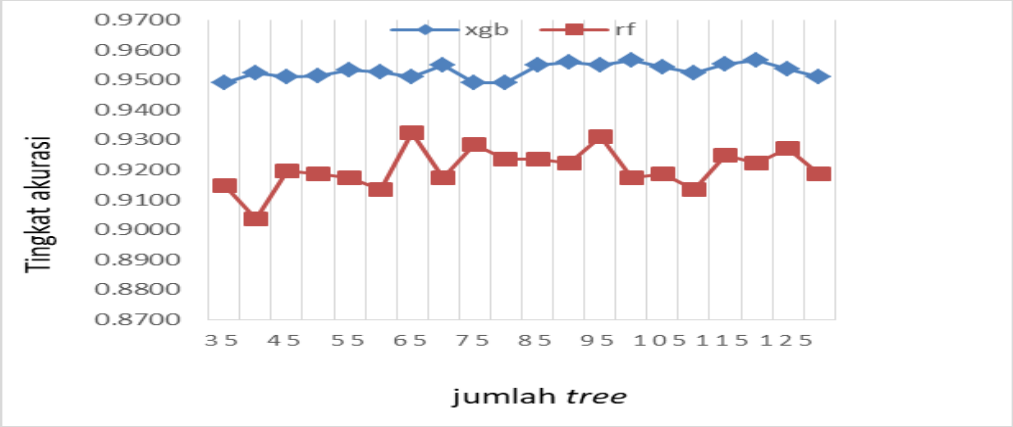


Figure 3. Accuracy level of both models by number of trees with default parameter

The resulted values show the accuracy level of each model built by using the default parameter with various combinations of tree number. The average level of accuracy of random forest is at 0.92 while xgboost is 0.95. The accuracy level of both methods to splice junction sample dataset is relatively high. Reconfiguration was done for the number of tree while no adjustment was made for other parameters, and accuracy value is estimated not to change significantly. To increase accuracy value, tuning hyperparameter on both methods was carried out

Optimization of Hyperparameter tuning by Grid Search

On this stage, analysis is conducted to obtain sequential patterns to be tested. Pattern in the form of grid allows the appropriate hyperparameter formulation for the appropriate accuracy level.

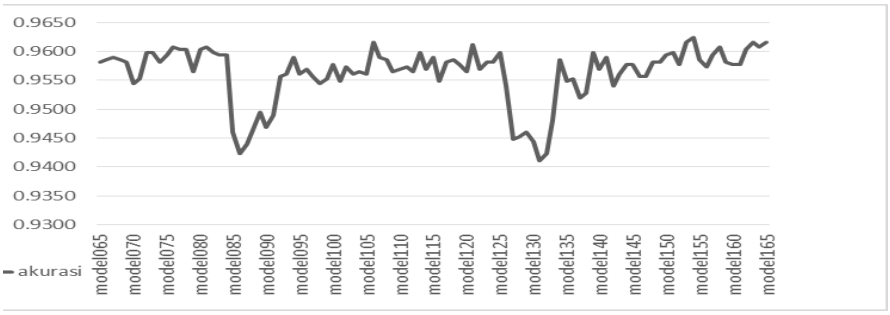
XGBoost Hyperparameter Tuning

Hyperparameter to be configured for xgboost are the depth of tree (max_depth), minimum weight of child (min_child_weight), subsample ratio for training process (subsample) and ratio subsample of column when building each tree (colsample_bytree). A default value is set for other *hyperparameters*. Other h*yperparameters* that can be adjusted include number of iteration (nround), regularization value (gamma) and *learning rate* (eta).

Hyperparameter search were conducted manually in 168 trials with various configurations. The best result obtained was at 96.24%. Hypermparameter configurations used are displayed in Table. 3.

Tabel 3. Xgboost *hyperparameter configuration*

No	Hyperparameter	Nilai	Mekanisme *tuning*
1	*nrounds*	80	manual
2	*eta*	0,2	manual
3	*gamma*	0	manual
4	*max_depth*	5	manual
5	*min_child_weight*	5	manual
6	*subsample*	0,4	manual
7	*colsample_bytree*	1	manual
8	*Boost_type*	gbtree	*fix*



Gambar 4. Akurasi model-model xgboost dengan berbagai konfigurasi parameter

Figure 4. displays test results on xgboost generated models. The graphic shows a dynamic move of accuracy level, inappropriate hyperparameter implementation resulted in prediction values that are far below accuracy values during test process by default value. Xgboost with more than five combinations of hyperparameters are fairly difficult to adjust on the hyperparameter configuration so that maximum accuracy value is obtained

Random Forest hyperparameter Tuning

The *Hyperparameter* configured in *random forest* are only the number of tree and number of features for sorting, so that the process to determine the hyperparameter becomes faster.

From Figure 5, it is seen that optimum values are generated by hyperparameters with ntree value of 905 with 5 variables mtry. The naming of each model in figure 5 refers to the hyperparameter configuration in terms of the values of m (mtry) and n (ntree).

Best Technique analysis

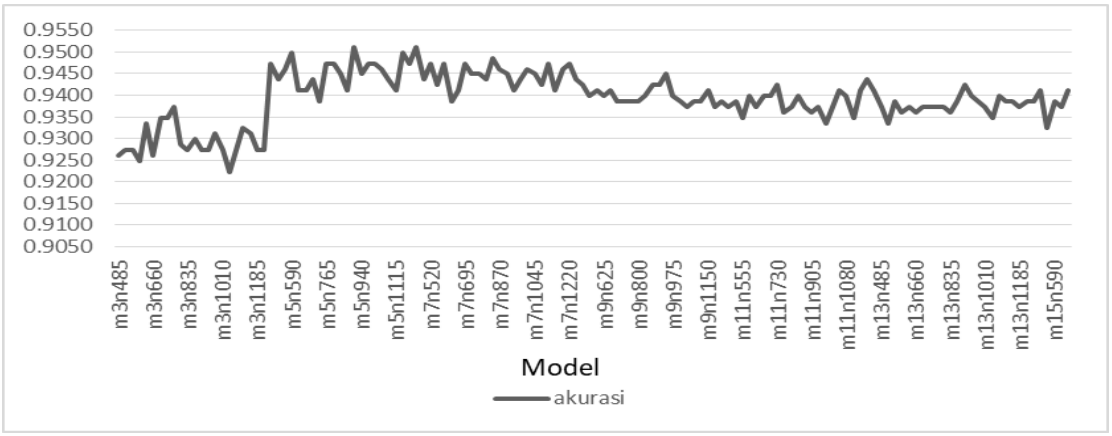


Figure 5. Accuracy of RF models built by various parameter configurations

From the testing, results of the comparison of accuracy levels of both methods both by default value and by tuning hyperparameter shown in Figure 6. From this figure, it can be concluded that xgboost method is superior to random forest. Even after random forest tuning is conducted, the level of accuracy obtained cannot exceed that of xgboost by default values.

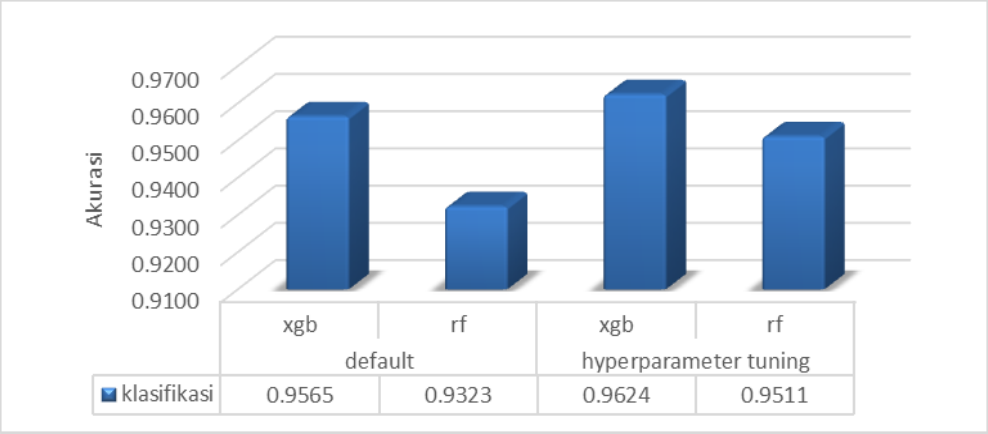


Figure 6. Best accuracy of built models.

Mechanism Comparison analysis

The bagging and boosting methods of the ensemble concept are different. Their general similarity is the use of more than one classifiers in their processes. Both methods have advantages and disadvantages. From this study, which uses small size dataset sample, it is indicated that xgboost is superior to that of random forest. Referring to several literature studies, the differentiation between the ensemble concepts of bagging and boosting is summarized in table 4.

Tabel 4. Analysis of the comparison of xgboost and rendomfest in this study

	XGBoost	**Random Forest**
Process mechanism	sequential	parallel
Number of *hyperparameter*	More than 5	Only 2
Training mchanism	Using all data with residue optimization	Menggunakan subsample secara acak
Ensemble mechanism	*boosting*	*bagging*
Use of a large number of *tree*	Tends to *overfit*	More *robust*
Types of *Decision tree*	*Shallow tree*	*Deep tree*

CONCLUSIONS

This study show that the ensemble methods of both boosting and bagging are able to handle classification in a good manner, when the hyperparameter is appropriately determined. The accuracy level of xgboost is overall superior. However, the drawback of xgboost is that its training process took more time to complete because within that process, the trees are built sequentially. The study also finds that it is more difficult to carry out hyperparameter tuning for xgboost. In addition, xgboost is more sensitive, so that when there is too much dirty data and too many outliers, overfitting may occur.

In random forest, training process of each tree is carried out independently, with random data sample. This randomization makes increase models' resistance and reduce overfitting of training data. The advantage of this model is the ease of parameter tuning compared to that of XGboost. The configuration process only requires two parameters, namely number of tree and number of features to be selected for each node. One of the disadvantages of the random forest method is the large number of tree built resulting in the longer process time for real time implementation.

Further researches are suggested to use more complex and massive size DNA sequence dataset in

order to find out the actual performance of XGBoost om DNA sequence pattern related to splice acceptor and donor. Outlier data may be removed so that models with more optimum value may be obtained.

Optimization may be performed with most ideal hyperparameter configuration search using random search. It is expected that hyperparameter values which are not included in the grid search pattern range can be found, so that configuration values can be used on models and possibly resulted in better accuracy.

REFERENCES

Barros, R. C., Basgalupp, M. P., de Carvalho, A. C., & Freitas, A. A. (2012, July). A hyper-heuristic evolutionary algorithm for automatically designing decision-tree algorithms. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation* (pp. 1237-1244). ACM.

Bonab, H. R., & Can, F. (2017). Less is more: a comprehensive framework for the number of components of ensemble classifiers. *arXiv preprint arXiv:1709.02925*.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).

Dietterich, T. G. (2000, June). Ensemble methods in machine learning. In *International workshop on multiple classifier systems* (pp. 1-15). Springer, Berlin, Heidelberg.

Farid, D. M., Zhang, L., Rahman, C. M., Hossain, M. A., & Strachan, R. (2014). Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. *Expert Systems with Applications*, *41*(4), 1937-1946.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

Libbrecht, M. W., & Noble, W. S. (2015). Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, *16*(6), 321.

Lichman, M. (2013). UCI machine learning repository.

Lo, C., Kakaradov, B., Lokshtanov, D., & Boucher, C. (2014). SeeSite: characterizing relationships between splice junctions and splicing enhancers. *IEEE/ACM transactions on computational biology and bioinformatics*, *11*(4), 648-656.

Sun, Z., Sang, L., Ju, L., & Zhu, H. (2008). A new method for splice site prediction based on the sequence patterns of splicing signals and regulatory elements. *Chinese Science Bulletin*, *53*(21), 3331.

Yang, P., Hwa Yang, Y., B Zhou, B., & Y Zomaya, A. (2010). A review of ensemble methods in bioinformatics. *Current Bioinformatics*, *5*(4), 296-308.