

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,200

Open access books available

169,000

International authors and editors

185M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



Chapter

# Identification of RNA Oligonucleotide and Protein Interactions Using Term Frequency Inverse Document Frequency and Random Forest

*Eugene Uwiragiye and Kristen L. Rhinehardt*

## Abstract

The interaction between protein and Ribonucleic Acid (RNA) plays crucial roles in many biological aspects such as gene expression, posttranscriptional regulation, and protein synthesis. However, the experimental screening of protein-RNA binding affinity is laborious and time-consuming, there is a pressing desire of accurate and reliable computational approaches. In this study, we proposed a novel method to predict that interaction based on both sequences of protein and RNA. The Random Forest was trained and tested on a combination of benchmark datasets and the term frequency–inverse document frequency method combined with XgBoost algorithm was used to extract useful information from sequences. The performance of our method was very impressive, and the accuracy was as high as 94%, the Area Under the Curve of 0.98 and the Matthew Correlation Coefficient (MCC) of 0.90. All these high metrics, especially the MCC, show that our method is robust enough to keep its performance on unseen datasets.

**Keywords:** protein, RNA, interaction, random forest, TFIDF, machine learning

## 1. Introduction

The protein-RNA pairs are highly involved in various regulatory processes. Finding the binding sites of the RNA-binding Proteins (RBP) is therefore an important research goal. Studies have shown that RBPs bind to RNA molecules by recognizing both sequences (sequence motifs) and secondary structure contexts (structure motifs) [1–4]. Some of them have been based on sequence-derived features such as amino acid composition, dipeptide composition, composition-transition-distribution of seven physicochemical properties, evolutionary information in terms of position-specific scoring matrices and functional domain composition [5–7]. Although progress has been made in the implementation of predictive methods for RBPs, insufficient attention has been paid to the development of predictive methods for RNA-protein

interactions (RPI). The history is brief, and there are not many existing computational tools because of the scarcity of available data [8].

The machine learning (ML) methods, which have become standard tools in many fields of science and engineering, are computationally efficient methods that employ computer science, artificial intelligence, computational statistics, and information theory to fit high-dimensional models to large amounts of data. The ML methods read in data points which are generated within some application domain and each data point is characterized by two properties, such as features (predictor variables) and labels (predicted variables). The machine learning algorithms aim at learning to predict the label of a data point based solely on the features of this data point or identify the pattern those data points if they are neither classified nor labeled. The ML algorithms applied to labeled data points is called supervised learning in contrast to unsupervised learning which does not require knowing the label value of any data point. The dataset we used in this research was tagged with known labels (binding pairs are labeled as positive while non-binding pairs are labeled as negative). While the principle behind supervised ML sounds trivial, the challenge of modern ML applications is the data points non-linearity and complexity. This research focuses on three supervised learning algorithms: Logistic Regression, Random Forest, and Multinomial Naïve Bayesian.

The logistic regression is a binary classification method that can be applied to data points with feature vector  $X \in R^n$  and binary labels  $y$ . These binary labels take on values from a label space that contains two different label values (most cases  $y = \{0,1\}$ ). The linear operator  $h(x) = w^T x$ , with  $w \in R^n$ , can take an arbitrary real random number and can predict the label  $y$  when compared to a given threshold. The data point with feature  $x$  would be classified as  $y = 1$  if the  $h(x) \geq 0$  and  $y = 0$  if the  $h(x) < 0$ . The multinomial naïve Bayesian is a simple but important probabilistic model which is defined by a function  $h$  from the feature space  $X$  to the label space  $Y$  ( $h : X \rightarrow Y$ ) such that the predicted value  $h(x), x \in X$ , agrees enough with the true value  $y \in Y$ . The random forest is a flowchart-like description of a function from the feature space to label space that maps the features to their respective labels. While a random forest can be applied to an arbitrary feature space, we will discuss it for a specific space later in this paper.

In 2011, Pancaldi and Bähler [8–10] predicted the RNA-binding proteins and messenger-RNA using two conventional machine learning classifiers: support vector machine (SVM) and random forest (RF), while Bellucci et al. developed an algorithm called catRAPID to facilitate the predictions of 592 RPIs from the Protein Database Bank (PDB). They used the physicochemical properties of sequences as features and found three most predictive features: secondary structure propensities, hydrogen bonding, and van der Waals [8, 11]. The two benchmark datasets, called RPI369 and RPI2241, were constructed from PRIDB (a database of protein-RNA interfaces) [8, 12, 13] and achieved remarkable prediction accuracies on these two datasets using Conjoint Triad Feature (CTF) and normalized 4-gram frequencies. In 2013, the CatRAPID Omics was generated as an improved CatRAPID that used the information on protein and RNA domains involved in macromolecular recognition [8, 14, 15]. Zhao Hui-Zhan et al. [8, 16] proposed a deep learning model to predict RPIs using bi-gram from Position Specific Scoring Matrix (PSSM) approaches to extract features from proteins, and k-mers approach combined with a stacked auto-encoder for RNAs feature extraction.

In 2015, Suresh et al. [8, 17] integrated sequence information and predicted structure together to produce an accurate prediction of non-coding RNA-protein pairs on a newly constructed dataset, called RPI1807. When tested on the RPI369 and RPI2241 datasets mentioned above, some improvements were achieved on prediction

accuracies. In 2017, Liu et al. proposed a semi-supervised method called LPI-NRLMF [18, 19] to predict lncRNA-protein interactions by neighborhood regularized logistic matrix factorization. One year later, Zhao et al. came up with IRWNRLPI method [20], integrating random walk and neighborhood regularized logistic matrix factorization for lncRNA-protein interactions prediction and LPI-BNPRA method using the bipartite network projection recommended algorithm to identify lncRNA-protein interactions. The last four semi-supervised methods and the BNPMMA method proposed by Chen et al. [21] in late 2018, performed well only on interactive pairs with a high predictive accuracy but weakly for non-interactive pairs. In 2018, Hu et al. proposed HLPI Ensemble method [22] for identifying lncRNA-protein interactions in human only, which integrated three common machine learning algorithms, SVM, RF and Extreme Gradient Boosting (XGB).

All the machine learning methods discussed above, use handcrafted features from proteins. In this study we proposed a new method, called TF-IDF borrowed from natural language processing, to extract features from RPI pairs. The TF-IDF standing for Term Frequency–Inverse Document Frequency takes as input a sequence of strings and transform it into a vector of numerical values.

## 2. Material and methodology

According to Hongchu Wang and Pengfei Wu in 2017 [8] there are 1973 RPI complexes available in the Protein Data Bank (PDB), which contains over 15,000 protein chains and more than 3000 RNA chains. However, according to research using high-throughput sequencing techniques (such as RNA-Seq), at least 30,000 lncRNAs were identified by 2013. In this study we combined the three different datasets; The RPI2241 dataset, containing 2241 RNA–protein pairs was extracted from PRIDB [13] and reconstructed by Wang in 2013, the RPI488, a non-redundant lncRPI dataset based on structural complexes which consists of 488 lncRNA-protein pairs, including 245 non-interacting pairs and 243 interacting pairs from Pan et al. [23, 24] and the RPI12737 dataset containing 12,737 experimentally validated RNA–protein pairs that extracted from NPInter v2.0 database [25]. This dataset contains the same number of non-interacting RNA–protein pairs (negative examples) as the number of interacting RNA–protein pairs. After the dataset combination, we cleaned the data by removing all pairs containing a non-amino acid character for proteins or a non-nucleotide for RNA. The difference between lengths of sequences could increase the sparsity of the TF-IDF data frame and affect the performance of our predictive model. The exploratory data analysis gave more details on the dataset (see **Table 1**). The first quartile of proteins lengths was 252 while the third quartile was 614, which means that the lengths of 50% of our combined dataset lie between 252 and 614. After all considerations, we decided to use this 50% of the dataset, containing 10,715 clean pairs, to train and test the predictive model.

### 2.1 Transformation of the sequence into text format

The biological sequences are sequences of successive letters without space with different lengths which are relevant to their biochemical structure and for their biological function. The bioinformaticians use the alignment process to arrange the primary structure of a protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences.

Dataset	Positive pairs	Negative pairs	#RNA	#Protein
RPI488	243	245	25	247
RPI2241	2241	2241*	841	2042
RPI12737	12,737	12737*	4636	449

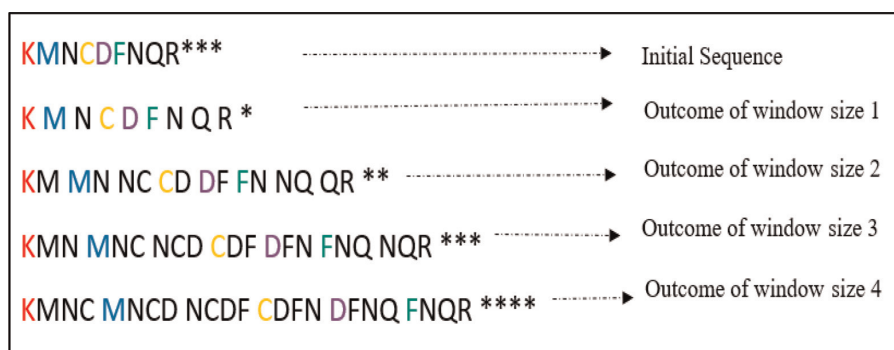
\*The number with star means that the negative pairs were not reported. All non-reported pairs are considered as negative.

**Table 1.**  
Description of different dataset used in random forest training.

This approach leaves a lot of holes in a sequence when a region from sequence of interest is not similar to the region of the other sequence. The alignment of multiple sequences is not as simple as it may seem at the first glance and the position feature of amino acids is threatened. Therefore, we propose a method which conserves the position feature of amino acids in the sequence by translating sequences into terms to apply the same representation technique for text data. The window with a subrange in the sequence that gives the best metrics was used and slipped through the given sequence with a fixed step, and each nucleotide (amino acid) segment was stored as a term. The shortest size of the sliding window that gave better metrics values was the size 3. As in the example below, the illustration of sequence transformation using a sliding window with size from one to four (**Figure 1**):

## 2.2 TF-IDF for feature engineering

The natural language processing has various types of approaches to transform the sequence of words into numerical values, such as the bag of words, words embedding, the term frequency inverse document frequency (TF-IDF), and so on. The TF-IDF measures the frequency of a term in a sequence which highly depends on the length of the sequence. The purpose of this method is to vectorize sequences [26–30]. To solve the sequence issue with the complicated alignment, TF-IDF method uses the combination of all possible terms in the dataset to have vectors of the same length with two extreme cases where TF value will be zero if the term does not appear in the dataset and 1 if all terms in the sequence are the same. The Term Frequency (TF) is used to measure how many times a term is present in a sequence and The Inverse Document Frequency assigns lower weight to frequent terms and assigns greater weight for the terms that are infrequent [31–33]. The TFIDF is the most widely used term weighting scheme. Yang and Huang [34] used it for calculating term weight according to the location and length of the



**Figure 1.**  
Illustration of a sequence to text format using a sliding window of different sizes.



keyword and Tian Xia and Yanmei Chai [35] implemented it by calculating distribution based on local term weighting and global term weighting to improve the efficiency of IR and TC systems and many researchers used the TF-IDF for feature engineering [36–38] to solve classification problems in reasonable time, efforts, and resources.

Assuming  $S$  a set of sequences:  $S = \{s: s \text{ is a sequence}\}$  and  $T$  a set of terms:  $T = \{t: t \text{ is a term}\}$ .

TF would be a function defined as follow:

$$TF : T * S \xrightarrow{TF} [0, 1] : (t : s) \xrightarrow{TF} TF(t, s) = \frac{\text{Number of appearance of } t}{\text{Number of terms in } s} \quad (1)$$

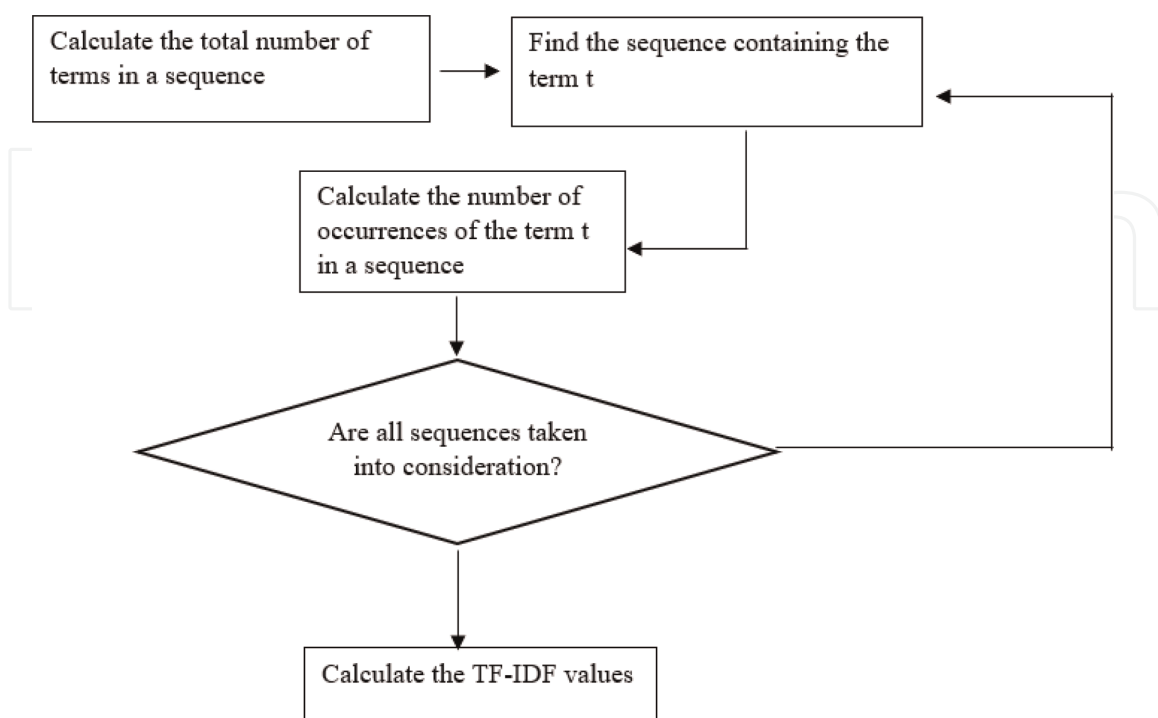
Where  $t$  is a given term in a sequence  $s$ . The IDF function or normalization function which calculates the importance of a sequence in the dataset will be defined as follow:

$$IDF(S_t) : S_T \xrightarrow{IDF} \mathbb{R} : S_t \xrightarrow{IDF} IDF(S_t) = \frac{N}{S_t} \quad (2)$$

Where  $S_T$  is the set of all sequences containing the term  $t$  and  $N$  is the number of all sequences in the dataset and  $s_t = |S_T|$ . Thereafter, the TF-IDF is the multiplicative value of  $TF(t,s)$  and  $IDF(s_t)$

$$TFIDF(t, s) = TF(t, s) * IDF(S_t) = \frac{N_t^s * N}{N_t * S_t} \quad (3)$$

Where  $N_t^s$  is the number appearances of term  $t$  in a sequence  $s$  and  $N_t$  is the number of sequences containing the term  $t$  (**Figure 2**).



**Figure 2.**  
 The term frequency-inverse document frequency flowchart.

## 2.3 Feature selection

The TF-IDF method vectorises the RNA and RBP sequences and transforms them into a 2D data frame with 10,715 rows and 7461 columns. In this situation, the dimensionality reduction is required. The XgBoost method, an optimized implementation of gradient boosted decision trees in python libraries, was used to estimate the importance of TF-values. That estimation consists in comparison of all attributes to each other, to rank them based on their contribution to the general classification. Extreme gradient boosting (XGBoost) is a new method that it can take weak feature classifiers and into one strong classifier [39] due to its gradient boosting algorithm, efficiency, flexibility, and portability [40, 41]. The XGBoost was used in the literature to discover and retain the features that highly impact the prediction [42–46] and was ten times less computationally expensive compared to other popular techniques [42].

## 2.4 Dataset balancing

In the 10,715 samples we have, 6333 were labeled as positive samples (interacting pairs) while other 4382 were labeled as negative samples. The 1951 samples of difference between two classes are not enormous. However, most machine learning algorithms do not work very well with such imbalanced datasets [31, 47, 48]. This is why we tried to train our model on unbalanced dataset and balance it thereafter. There are several techniques to balance datasets [32] but we chose to use two of them: Random Oversampling by using the bootstrapping method to increase the size of the minority class, and Under sampling that applies a nearest-neighbors algorithm [48] and “edit” the dataset by removing samples which do not agree “enough” with their neighborhood.

## 3. Predictive model: random Forest

The prediction of RPIs was done after training and testing the Random Forest among other classifiers. The RF is a supervised machine learning algorithm that is constructed from decision tree algorithms developed by Tin Kam Ho in 1995 [33, 34] and used to solve classification and regression problems. The random forest establishes the result according to the mean predictions of all the decision trees. A decision tree consists of decision nodes, leaf nodes, and a root node. The algorithm behind the decision tree divides the training set into branches, which further split into new branches until a leaf node is attained (a leaf node cannot be splitted into other branches). This sequence of branches uses the Classification And Regression Tree (CART) methodology combined to the resampling with replacement [25]. The random forest has multiple parameters that can be optimized by most of them were kept by default. Among the parameters the criterion Gini and the minimum of sample required to split fixed at two trees and hundred branches were chosen for better results.

### 3.1 Classification trees (Forest)

A decision tree is a way of representing knowledge obtained in the inductive learning process. The space is split using a set of conditions, and the resulting structure is the tree. Assuming we have  $n$  pairs and TF-values vectors  $\{X_i\}_{i=1}^n$  with outcomes  $y_i$ , our dataset can be presented as follow:

$$\text{Dataset} = \{(X_1, y_1), (X_2, y_2), \dots, (X_n, y_n)\} \quad (4)$$

Each TF-value vector is  $X_k = (X_{k1}, X_{k2}, \dots, X_{kd})$  and  $d$  is the number of TF-values from RNA and RBP altogether.

The decision tree is defined as binary process where a decision is made based on whether the TF-value  $X_i$  is inferior to a threshold  $t$  or not. This threshold depends on the node at which the decision is made. The top node contains all examples  $(X_n, y_n)$ , and these examples are subdivided into children nodes according to the possibility of classification at that node. The subdivision of examples continues until every node at the bottom has examples which are in one class only.

### 3.2 The Gini criterion

The Random Forest as a python implementation of the scikit-learn library, this is made by the parameter 'criterion'. This parameter is the function used to measure the quality of a split and it allows users to choose between 'Gini', or 'entropy'. We preferred the Gini criterion because computationally, entropy is more complex since it makes use of logarithms and consequently, the calculation of the Gini Index will be faster. The Gini criterion is used to measure the diversity at each tree node when the TF-value and optimal threshold are chosen. Assuming the set of all examples is  $S$  and the set of examples at the node  $j$  is  $S_j$ , then  $S$  is a partition of children node sets, i.e.:

$$S = \bigcup_1^l S_j \text{ where } l \text{ is the number of children nodes}$$

Each sample  $S_j$  is portioned into two classes  $C_1$  = interacting pair and  $C_2$  = non-interacting pair. The proportion of a sample  $S_j$  in the set of all examples and the proportion of  $S_j$  with a class  $C_i$  are respectively defined as follow:

$$P(S_j) = \frac{|S_j|}{|S|}$$

$$P(C_i|S_j) = \frac{|S_j \cap C_i|}{S} \quad (5)$$

The Gini criterion is the variation  $g(S_j)$  in the set  $S_j$  defined as follow:

$$g(S_j) = \sum_1^1 P(C_i|S_j) (1 - P(C_i|S_j)) \quad (6)$$

The variation  $g(S_j)$  reaches the maximum when the set  $S_j$  is equally divided in the class  $C_i$  and the minimum when the set  $S_j$  is just made by one of the two classes. The variation the full subdivision  $S_j$  (known as Gini Index) is defined as the weighted sum of their respective proportions in the set of all examples.

$$\text{Gini Index} = P(S_1)g(S_1) + P(S_2)g(S_2) + \dots P(S_l)g(S_l) \quad (7)$$

### 3.3 The random vector

A random vector is defined as an array  $X$  of random variables defined on the same probability space. In this study the array is the TF-values vectors



$$X = (X_1, X_2 \dots X_d) \text{ where } X_i \text{ are column vectors} \quad (8)$$

The random  $y = \{y_1, y_2, \dots y_d\}$  with  $y_i \in \{0,1\}$  is the classification of examples where 1 represent a protein-RNA interaction (RPI) while 0 represent a non-interaction. The model vector  $(X,y)$  is defined on the same probability space as the random vector  $X$ .

The goal of this predictive model is to build a classifier which predict the random vector  $y$  (classes) from random vector  $X$  (TF-values) based on the examples in the dataset from paragraph 3.1. This classifier is based on a family of classification trees and the ensemble of those trees is called Random Forest.

### 3.4 Ten-fold cross-validation method

The cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called  $k$  that refers to the number of groups that a given dataset is to be split into, and it is called  $k$ -fold cross-validation ( $k = 10$  for this study). The 80% was used for the 10-fold cross validation, randomly shuffled and split into 10 groups. Among the 10 groups, only one group was kept as validation data to test the model and the remaining 9 sub-samples were used as training data. Importantly, each observation in the validation set is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set 1 time and used to train the model 9 times. The 10 results were then averaged to produce a single estimate by summarizing the mean of the model scores. The metrics we used to evaluate the model performance are Accuracy, Specificity, Sensitivity and MCC (Matthews Correlation Coefficient)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + TN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{MCC} = \frac{TP * TN - FP * FN}{\sqrt{(TP + TN)(FP + FN)(TP + FN)(TP + FP)}} \quad (9)$$

Where TP, FP, TN, and FN stand for True Positive, False Positive, True Negative and False Negative respectively.

### 3.5 Independent test

The remaining 20% of the dataset was used to test the classifier performance to the unseen data. This test dataset was completely independent of the data sample used in 10-fold cross validation. The goal was to train the Random Forest with parameters having the best performance on new data.

## 4. Results and discussion

We have applied the sliding window approach to transform RNA and protein sequences into text format using different window's sizes starting from size 2. We

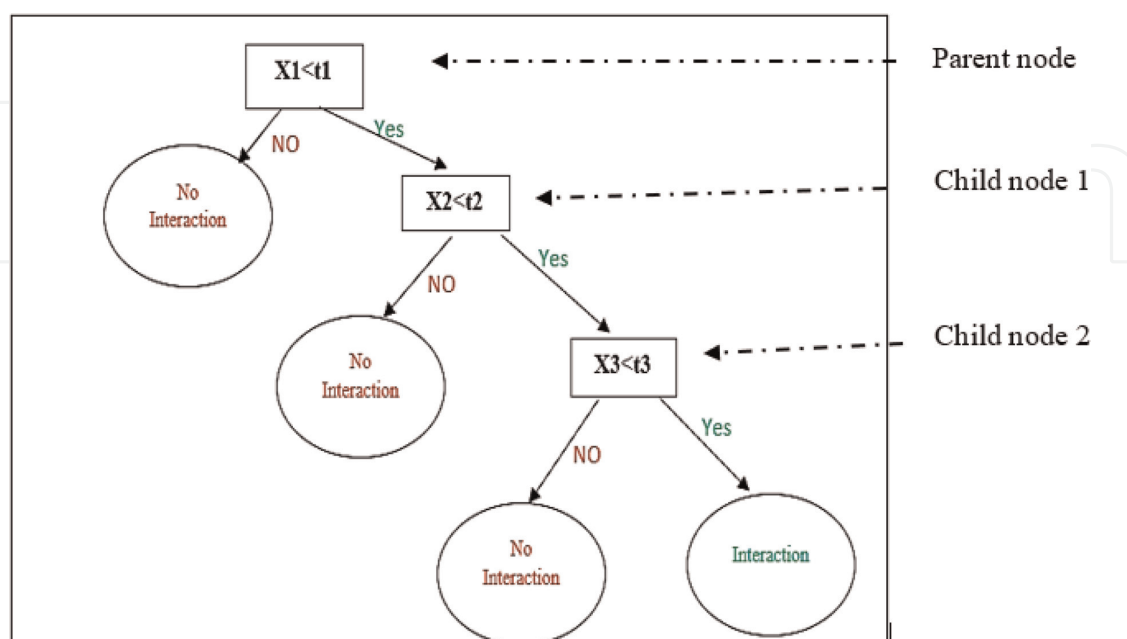
constated that there was not much difference between the model performance when using a sliding window size of 3,4 or 5 and the performance started decreasing at window size of 6. Therefore, we chose the window's size 3 because it gives the best results in less time. After we applied the TF-IDF to the dataset we got a data frame of 10,715 rows and 7461 attributes. The Random Forest applied to this dataset gave a good performance with a scope of improvement because all 7461 features do not have the same importance in the prediction. We applied the XgBoost algorithm to select the best features. The best threshold showed that 232 features contribute to the prediction at 0.2% at least. The performances of different classifiers are summarized in **Table 2**.

The receiver operating characteristic (ROC) curves of the three classifiers confirms our preference of the Random Forest to other classifiers. The Area under the curve is 0.98, 0.95 and 0.93 for Random Forest, Logistic Regression and Multinomial Naïve Bayesian respectively (**Figure 3**). Sometimes, one algorithm can overperform other algorithms for one metric measure and loses for other metrics. But in this study, the

Classifiers	10-Fold cross validation				Independent test			
	<i>Spe</i>	<i>Sen</i>	<i>Acc</i>	<i>MCC</i>	<i>Spec</i>	<i>Sen</i>	<i>Acc</i>	<i>MCC</i>
RF	0.96	0.94	0.95	0.92	0.95	0.94	0.94	0.90
LR	0.96	0.89	0.92	0.84	0.93	0.89	0.91	0.83
MNB	0.95	0.89	0.92	0.84	0.93	0.88	0.91	0.82

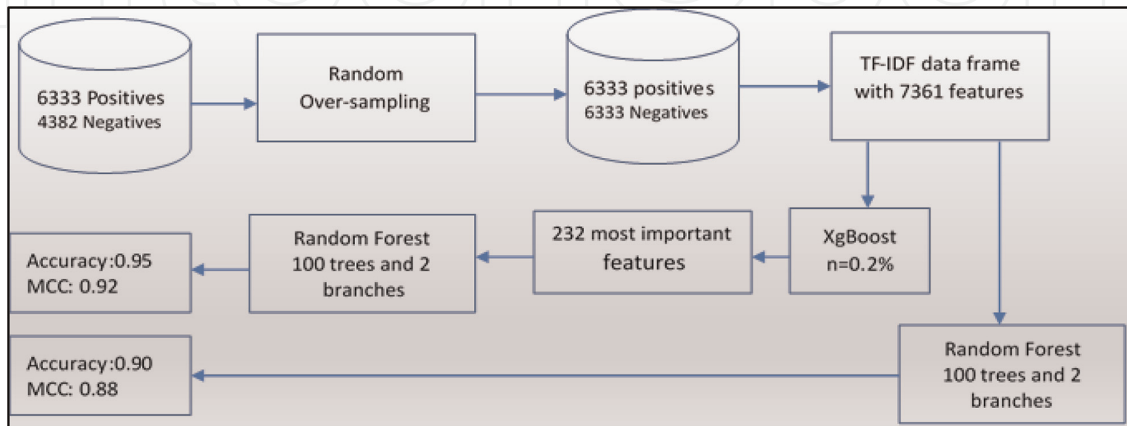
RF = Random Forest; LR = Logistic Regression; MNB = Multinomial Naïve Bayesian.

**Table 2.**  
 Comparative summary of three different predictive models.

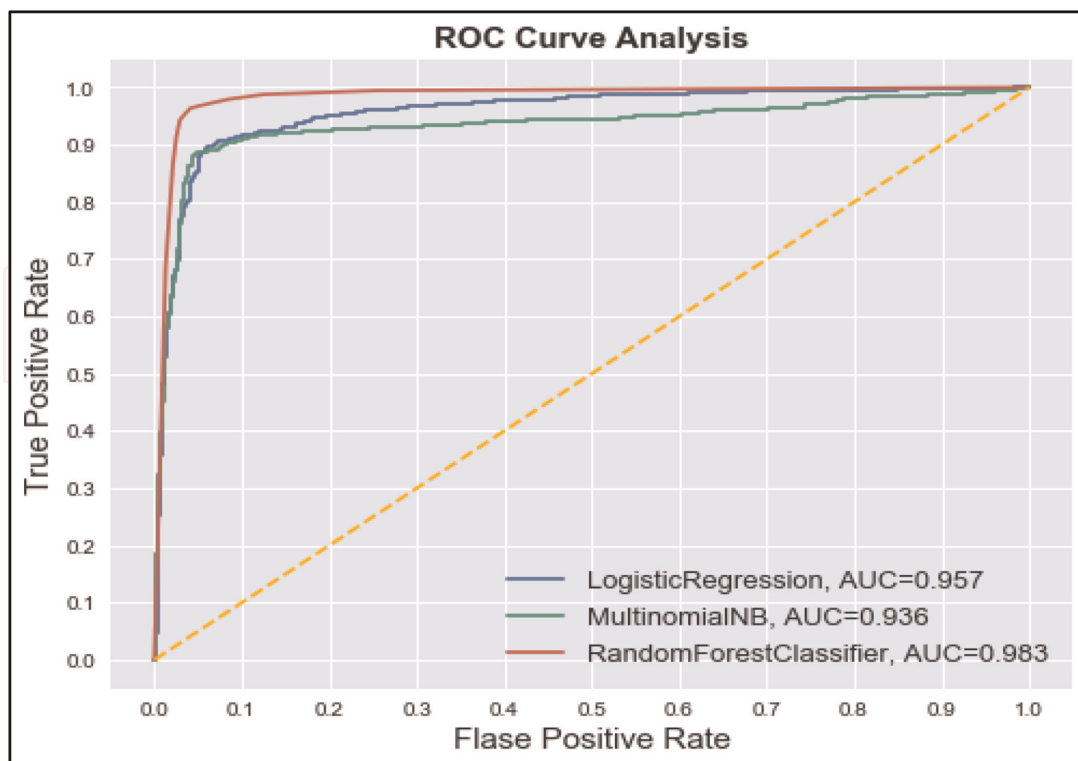


**Figure 3.**  
 Illustration of classification trees with three nodes. The thresholds  $t_i$  depends on each note and are learned during the training process.

Random Forest overperformed other two classifiers in all metrics and more importantly for Matthew Correlation Coefficient (MCC) because it is an ensemble-based algorithm using the resampling with replacement method to reduce variance. This method makes that the Random Forest takes a lot of time to be trained but it is worth it because: a tree-based learning algorithm, on large datasets, allows to quantitative and qualitative input variables, can be immune to redundant variables or variables with high correlation which may lead to overfitting in other learning algorithms and has few parameters to tune (Figures 4 and 5).



**Figure 4.**  
A systematic review of imbalanced data challenges and dimensionality reduction.



**Figure 5.**  
Representation of ROC of the AUC for three classifiers showing that the random Forest curve is higher than other classifiers.

## 5. Conclusions

The Term Frequency Inverse Document Frequency borrowed from natural language processing was combined with the sliding window to transform the RNA and protein sequences into a data frame of numerical values and 232 most contributing TF-values were selected using the XgBoost feature importance. Based on these features, we trained the Random Forest classifier on 10,132 samples and tested it on 2534 remaining samples. The results in the **Table 2** show that the Random Forest overperformed all other predictive models that we trained on this dataset for comparison such as Logistic Regression and Multinomial Naïve Bayesian. The highest AUC for the Random Forest, combined with the high specificity and sensitivity, provides an indication of its ability to correctly predict all classes in large datasets. The Random Forest is computationally expensive, but there is a significant performance difference compared to other classifiers which is worth the training time.

## Acknowledgements

This study was supported by Visualization and Computation Advancing Research (ViCAR) Center and funded by National Science Fund (NSF).


## Author details

Eugene Uwiragiye and Kristen L. Rhinehardt\*  
Computational Data Science and Engineering, North Carolina Agricultural  
and Technical State University, Greensboro, NC, United States of America

\*Address all correspondence to: [klrhineh@ncat.edu](mailto:klrhineh@ncat.edu)

## IntechOpen

---

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

## References

- [1] Jain DS, Gupte SR, Aduri R. A data driven model for predicting rna-protein interactions based on gradient boosting machine. *Scientific Reports*. 2018;**8**(1):1-10
- [2] Licatalosi DD, Darnell RB. RNA processing and its regulation: Global insights into biological networks. *Nature Reviews Genetics*. 2010;**11**(1):75-87
- [3] Kishore S, Lubner S, Zavolan M. Deciphering the role of RNA-binding proteins in the post-transcriptional control of gene expression. *Briefings in Functional Genomics*. 2010;**9**(5-6):391-404
- [4] Beckmann BM et al. The RNA-binding proteomes from yeast to man harbour conserved enigmRBPs. *Nature Communications*. 2015;**6**(1):1-9
- [5] Allers J, Shamoo Y. Structure-based analysis of protein-RNA interactions using the program ENTANGLE. *Journal of Molecular Biology*. 2001;**311**(1):75-86
- [6] Terribilini M et al. Prediction of RNA binding sites in proteins from amino acid sequence. *RNA*. 2006;**12**(8):1450-1462
- [7] Kim OT, Yura K, Go N. Amino acid residue doublet propensity in the protein-RNA interface and its application to RNA interface prediction. *Nucleic Acids Research*. 2006;**34**(22):6450-6460
- [8] Wang H, Wu P. Prediction of RNA-protein interactions using conjoint triad feature and chaos game representation. *Bioengineered*. 2018;**9**(1):242-251
- [9] Mercer TR, Mattick JS. Structure and function of long noncoding RNAs in epigenetic regulation. *Nature Structural & Molecular Biology*. 2013;**20**(3):300-307
- [10] Pancaldi V, Bähler J. In silico characterization and prediction of global protein-mRNA interactions in yeast. *Nucleic Acids Research*. 2011;**39**(14):5826-5836
- [11] Bellucci M et al. Predicting protein associations with long noncoding RNAs. *Nature Methods*. 2011;**8**(6):444-445
- [12] Muppurala UK, Honavar VG, Dobbs D. Predicting RNA-protein interactions using only sequence information. *BMC Bioinformatics*. 2011;**12**(1):1-11
- [13] Lewis BA et al. PRIDB: A protein-RNA interface database. *Nucleic Acids Research*. 2010;**39**(suppl\_1):D277-D282
- [14] Agostini F et al. Cat RAPID omics: A web server for large-scale prediction of protein-RNA interactions. *Bioinformatics*. 2013;**29**(22):2928-2930
- [15] Agostini F et al. X-inactivation: Quantitative predictions of protein interactions in the Xist network. *Nucleic Acids Research*. 2013;**41**(1):e31-e31
- [16] Zhan Z-H et al. BGFE: A deep learning model for ncRNA-protein interaction predictions based on improved sequence information. *International Journal of Molecular Sciences*. 2019;**20**(4):978
- [17] Suresh V et al. RPI-Pred: Predicting ncRNA-protein interaction using sequence and structural information. *Nucleic Acids Research*. 2015;**43**(3):1370-1379
- [18] Liu H et al. LPI-NRLMF: lncRNA-protein interaction prediction by neighborhood regularized logistic matrix factorization. *Oncotarget*. 2017;**8**(61):103975



- [19] Cheng S et al. DM-RPIs: Predicting ncRNA-protein interactions using stacked ensembling strategy. *Computational Biology and Chemistry*. 2019;**83**:107088
- [20] Zhao Q et al. IRWNRLPI: Integrating random walk and neighborhood regularized logistic matrix factorization for lncRNA-protein interaction prediction. *Frontiers in Genetics*. 2018;**9**:239
- [21] Chen X et al. BNPMDA: Bipartite network projection for MiRNA-disease association prediction. *Bioinformatics*. 2018;**34**(18):3178-3186
- [22] Hu H et al. HLPI-ensemble: Prediction of human lncRNA-protein interactions based on ensemble strategy. *RNA Biology*. 2018;**15**(6):797-806
- [23] Pan X et al. IPMiner: hidden ncRNA-protein interaction sequential pattern mining with stacked autoencoder for accurate computational prediction. *BMC Genomics*. 2016;**17**(1):1-14
- [24] Zhan Z-H et al. Accurate prediction of ncRNA-protein interactions from the integration of sequence and evolutionary information. *Frontiers in Genetics*. 2018;**9**:458
- [25] Yuan J et al. NPInter v2. 0: An updated database of ncRNA interactions. *Nucleic Acids Research*. 2014;**42**(D1): D104-D108
- [26] Trstenjak B, Mikac S, Donko D. KNN with TF-IDF based framework for text categorization. *Procedia Engineering*. 2014;**69**:1356-1364
- [27] Joachims T. Text categorization with support vector machines: Learning with many relevant features. In: *European Conference on Machine Learning*. Berlin, Heidelberg: Springer; 1998
- [28] Yang Y, Liu X. A re-examination of text categorization methods. In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1999
- [29] Soucy P, Mineau GW. Beyond TFIDF weighting for text categorization in the vector space model. In: *International Joint Conferences on Artificial Intelligence Organization*. Vol. 5. 2005
- [30] Xu G et al. Improved TFIDF weighting for imbalanced biomedical text classification. *Energy Procedia*. 2011;**11**:2360-2367
- [31] Beckmann M, Ebecken NF, de Lima BSP. A KNN undersampling approach for data balancing. *Journal of Intelligent Learning Systems and Applications*. 2015;**7**(04):104
- [32] Santos MS et al. A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients. *Journal of Biomedical Informatics*. 2015;**58**:49-59
- [33] Li B-Q et al. Prediction of protein-protein interaction sites by random forest algorithm with mRMR and IFS. *PLoS One*. 2012;**7**(8):e43927
- [34] Zhu C, Cheng G, Wang K. Big data analytics for program popularity prediction in broadcast TV industries. *IEEE Access*. 2017;**5**:24593-24601
- [35] Tian, X., and W. Tong. An improvement to tf: Term distribution-based term weight algorithm. 2010 *Second International Conference on Networks Security, Wireless Communications and Trusted Computing*. 2010. IEEE
- [36] Liu L, Peng T. Clustering-based method for positive and unlabeled text

categorization enhanced by improved TFIDF. *Journal Information Science Engineering*. 2014;**30**(5):1463-1481

[37] Qu S, Wang S, Zou Y. Improvement of text feature selection method based on tfidf. In: 2008 International Seminar on Future Information Technology and Management Engineering. IEEE; 2008

[38] Goswami P, Kamath V. The DF-ICF algorithm-modified TF-IDF. *International Journal of Computer Applications*. 2014;**93**(13)

[39] Li D et al. Feature selection and model fusion approach for predicting urban macro travel time. *Mathematical Problems in Engineering*. 2020;**2020**

[40] Brownlee J. XGBoost With python: Gradient boosted trees with XGBoost and scikit-learn. *Machine Learning Mastery*; 2016

[41] Chang W et al. A machine-learning-based prediction method for hypertension outcomes based on medical data. *Diagnostics*. 2019;**9**(4):178

[42] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*. 2016

[43] He X et al. Practical lessons from predicting clicks on ads at facebook. In: *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. 2014

[44] Pal A, Shrivastava N, Tripathi P. Comparison of Classification Algorithms Using Machine Learning. 2019

[45] Horrell M. Wide Boosting. arXiv preprint arXiv:2007.09855, 2020

[46] Bennett J, Lanning S. The netflix prize. In: *Proceedings of KDD Cup and Workshop*. New York; 2007

[47] Domingues I et al. Evaluation of oversampling data balancing techniques in the context of ordinal classification. In: 2018 International Joint Conference on Neural Networks (IJCNN). IEEE; 2018

[48] Rodríguez JP, Corrales DC, Corrales JC. A process for increasing the samples of coffee rust through machine learning methods. *International Journal of Agricultural and Environmental Information Systems (IJAEIS)*. 2018;**9**(2):32-52