Computational Biology and Chemistry 92 (2021) 107489

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/cbac

Research Article

SubFeat: Feature subspacing ensemble classifier for function prediction of DNA, RNA and protein sequences

H.M.Fazlul Haque [1], Muhammod Rafsanjani [1], Fariha Arifin [1], Sheikh Adilina, Swakkhar Shatabda *

Department of Computer Science and Engineering, United International University, United City, Madani Avenue, Badda, Dhaka 1212, Bangladesh

ARTICLE INFO

Keywords:
Feature subspacing
Ensemble classifier
Biological entities
Machine learning
Classification

ABSTRACT

The information of a cell is primarily contained in deoxyribonucleic acid (DNA). There is a flow of DNA information to protein sequences via ribonucleic acids (RNA) through transcription and translation. These entities are vital for the genetic process. Recent epigenetics developments also show the importance of the genetic material and knowledge of their attributes and functions. However, the growth in these entities' available features or functionalities is still slow due to the time-consuming and expensive in vitro experimental methods. In this paper, we have proposed an ensemble classification algorithm called SubFeat to predict biological entities' functionalities from different types of datasets. Our model uses a feature subspace-based novel ensemble method. It divides the feature space into sub-spaces, which are then passed to learn individual classifier models. The ensemble is built on these base classifiers that use a weighted majority voting mechanism. SubFeat tested on four datasets comprising two DNA, one RNA, and one protein dataset, and it outperformed all the existing single classifiers and the ensemble classifiers. SubFeat is made available as a Python-based tool. We have made the package SubFeat available online along with a user manual. It is freely accessible from here: https://github.com/fazlulhaquejony/SubFeat.

1. Introduction

With the advent of modern sequencing machines and techniques, there had been tremendous growth in known sequences. DNA, RNA, and proteins are of primary interest. They are involved in all information flow and even in epigenetics. A huge number of sequences and their attributes and properties are vital to understand cell organisms. Among these are secondary structure (Singh et al., 2019), gene-coding markers (Amin et al., 2019; Rahman et al., 2019b), anti-cancer properties (Gabernet et al., 2019), editing (Choyon et al., 2020), binding(Zaman et al., 2017; Chowdhury et al., 2017), post-translational modifications (Islam et al., 2018; Ahmad et al., 2020; Rashid et al., 2020), sub-cellular localization (Shatabda et al., 2017), methylation (Bell et al., 2019), and many other important process and functions that regulates almost all the processes within the cell organism. However, these techniques are time-consuming and expensive.

There has been growth in developing computational and knowledge-based methods to predict the sequences' attributes and functions (Liu, 2019; Chen et al., 2020; Muhammod et al., 2019; Kaushik et al., 2020; Xu et al., 2020). One of the key advantages of the knowledge based methods is that they often provide further insights to the patterns that are discoverable using fast computational facilities available and even with relatively small amount of data knowledge transfers and deep learning have also been possible from one problem to another (Namuduri et al., 2019; Zhou et al., 2020; Amin et al., 2019; Luo et al., 2019). One of the common approaches in the literature is to formulate the prediction task as a supervised learning problem: binary, multi-class, multi-label (Uddin et al., 2018; Taherzadeh et al., 2016). A number of successful classifiers have been used, single classifiers like support vector machines (SVM) (Uddin et al., 2018), K-nearest neighbors (KNN) (Ning et al., 2019), Decision Trees (DT) (Turan and Sehirli, 2017), Naive Bayes (NB) (Adilina et al., 2019), logistic regression (LR) (Ntranos et al., 2019); ensemble methods like AdaBoost (Rayhan et al., 2017), Random Forest (Li et al., 2018) have been applied to solve these problems. However, no single method seems to be performing well over other methods, and there is scope to develop new techniques.

* Corresponding author.
E-mail address: swakkhar@cse.uiu.ac.bd (S. Shatabda).
[1] Joint first authors.

https://doi.org/10.1016/j.compbiolchem.2021.107489
Received 21 August 2020; Received in revised form 7 March 2021; Accepted 19 April 2021
Available online 24 April 2021

One of the essential factors in building a successful machine learning-based method is the dataset's representation. In this case, it's how the sequences of DNA, RNA, and proteins are converted to a vector representation. Usually, ensemble methods are found to provide superior performances, provided that they utilize the underlying feature space properly. AdaBoost iteratively learns using weak classifiers. However, the algorithm does not exploit or consider the underlying feature space. On the other hand, Random Forest unexpectedly samples the features. From the point of view of the biological domain, it has been often seen that in many cases, the components are grouped into several sub-groups based on their respective generating techniques and sometimes the subgroups to share essential knowledge. Our main idea in this work is to utilize this property of the feature space.

In this paper, we present an ensemble method called *SubFeat*. *SubFeat* divides the full feature space into overlapping or non-overlapping subspaces and learns base classifiers (without ensemble classifier) or their mix on the sub-spaces, and the ensemble is created using a voting technique. It is much similar to Random Forest or Ensemble Voting process in how it uses the feature space and the voting mechanism. However, the approach taken to divide the subspace is unique here. We have tested the problem to four problems related to DNA, RNA, and proteins: DNA-binding proteins prediction using protein sequences, A-to-I editing prediction of RNA sequences, and promoter, and recombination hotspot prediction of DNA sequences. The datasets used in work are all standard benchmark datasets. SubFeat is a comprehensive Python-based tool that works with a limited features group. It provides the overlapping feature option that users can customize for their research purposes. It merely concentrates on overlapping feature hypothesis rather than feature variety of feature generation; the model outperformed after the rigorous experiment. SubFeat chooses two well-established feature groups: 'k-mer' and the 'gapped k-mer'. The SubFeat tool extended some of the functionality of the PyFeat tool (Muhammod et al., 2019). But the PyFeat tool predominantly focuses on numerical feature generation based on biological traits, it also provides classification tasks. 'Gapped k-mer' and 'k-mer' are well-known for the core feature selection as we have concentrated more on sub-feature base methods. We merely focus on core feature selection rather than its variation and biological attributes. We observed our experiment after full features space also did not perform well; on the other hand, when we divided the parts into subspaces and randomly selected the features, the performance enhances drastically (Table 4–7).

The experimental results show the superiority of the proposed method, *SubFeat* over several single classifiers and ensembles. We have made the methodology available as a Python package freely available and usable from: https://github.com/fazlulhaquejony/SubFeat.

2. Materials and methods

The basic idea of the ensemble method, *SubFeat* is given in Fig. 1. In this paper, we have divided the feature space into three sub-spaces. Each was then trained using a base classifier, and the final prediction is made based on the weighted majority voting of the sub-classifiers. The framework can utilize the possible overlap or non-overlap among the feature spaces, which are given in Figs. 2 and 3 .

In this section, we provide the details of our methods and materials. The section starts with a description of the datasets and the problems that were selected for experiments. A very brief literature review from the computational point of view is also provided for each problem. After that, we describe our feature representation for each of the problems. The ensemble is presented next with the choice of the algorithms in brief. We also describe the performance evaluation techniques used for the work.

2.1. Datasets

For this work, we have considered four problems: prediction of DNA recombination hotspots, prediction of promoter sequences in DNA, RNA A-to-I editing prediction, and prognosis of DNA binding proteins. Thus we have incorporated three types of sequences: DNA, RNA, and proteins. This section provides a description of the dataset collection and a brief literature review of the state-of-the-art methods of each of the problems. In supervised machine learning, a dataset is generally composed of positive and negative samples:

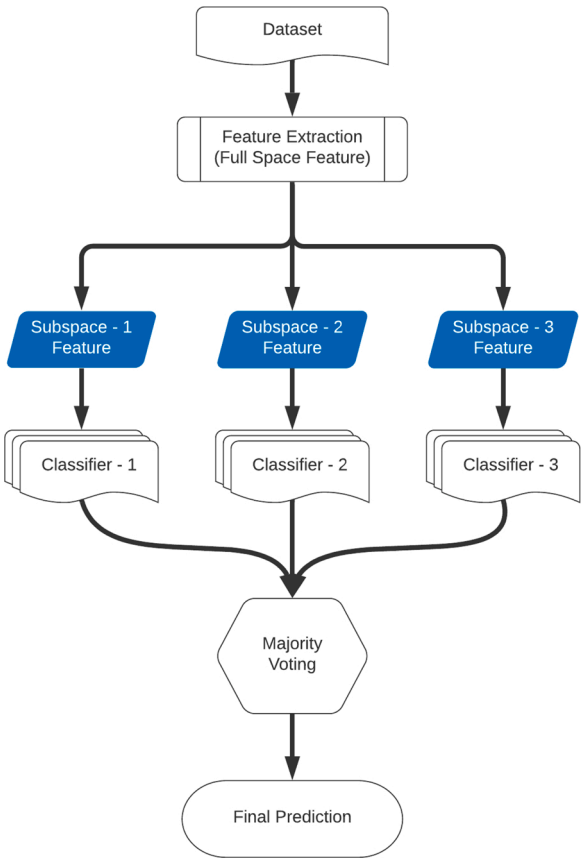

Fig. 1. Block diagram for ensemble classifier.

$$\mathbb{S} = \mathbb{S}^{+} \cup \mathbb{S}^{-} \tag{1}$$

Here, $\mathbb{S}^{+}$ denotes the set of positive instances, and $\mathbb{S}^{-}$ denotes the set of negative examples. We have selected an almost balanced dataset of examples where positive and negative classes are approximately equally distributed in our experiments. We have reduced the redundant sequence using the CD-HIT tool (Fu et al., 2012). We used four well-established datasets in our experiments, whereas two DNA, an RNA, and a protein dataset. A summary of the datasets used in this paper is given in Table 1.

2.1.1. Recombination hotspot

Hotspots are regions in the genome where meiotic recombination rates are much higher compared to the cold spots. DNA binding arrays are used *in vitro* to find recombination hot spots (Baudat et al., 2010; Jani et al., 2018). The dataset that we consider in this paper was originally curated by Jiang et al. (2007a). Recently, a good number of machine learning-based algorithms and methods (Al Maruf and Shatabda, 2019; Jani et al., 2018) as well as ensemble-based methods (Liu et al., 2017) are being proposed in the literature to solve the problem computationally. By using DNA microarray at the single-gene resolution, the relative recombination rates for the yeast *Saccharomyces cerevisiae* loci have been estimated by Rimmer et al. (2014). The hybridization ratio of P2/P1 estimated the relative strength of recombination. The ratio of hybridization to a DSB-enriched probe (P2) to a

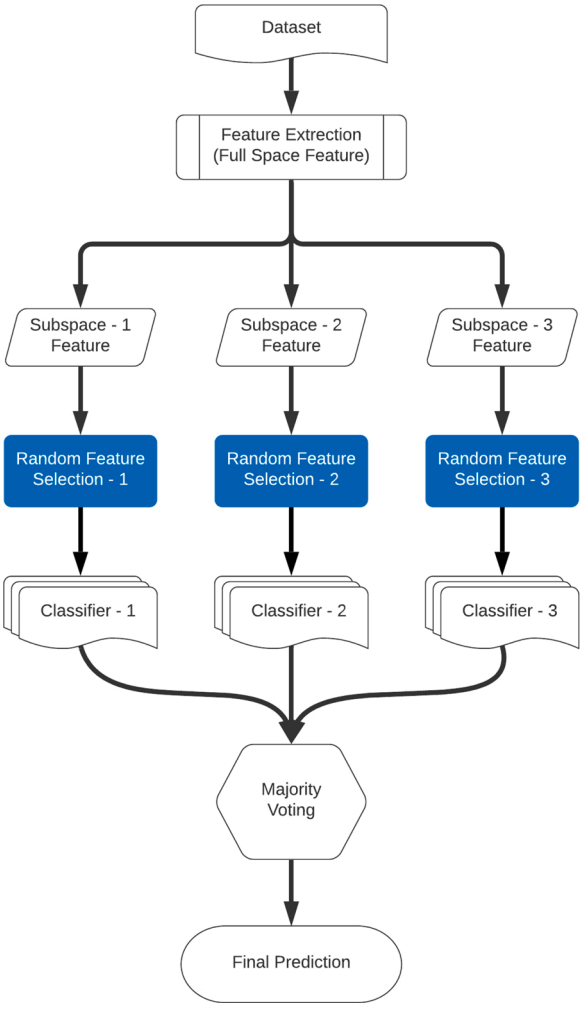

Fig. 2. Block diagram for non-overlap ensemble classifier.

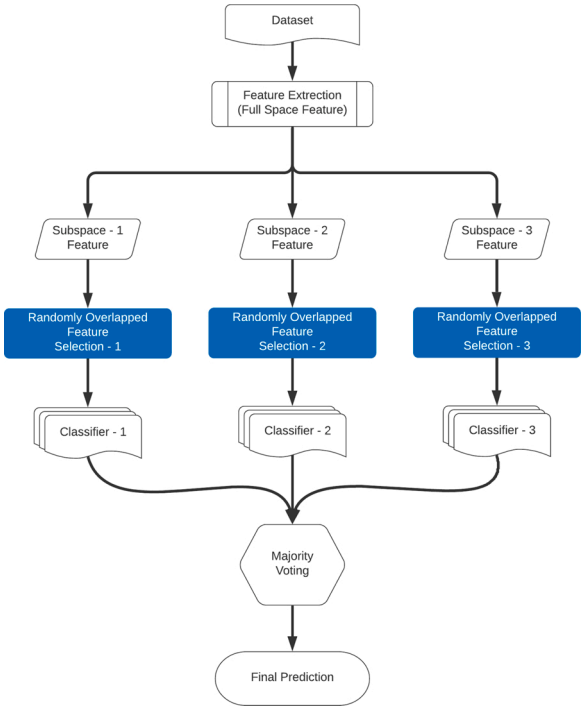

Fig. 3. Block diagram for overlap ensemble classifier.

total genomics probe (P1) was measured to estimate the DSBs formation adjacent to each ORF. They take the median value as the relative recombination rate of each sequence, in their article. For each of the 6200 genes, the experiments were repeated seven times. Here the sequence was excluded for an array value that was missing. Thus a total of 5266 sequences were culled. For relative hybridization ratio $\geq$1.5 are defined as hotspots, and relative hybridization ratio $<$0.82 are defined as coldspots of those sequences. By this procedure, at last, 490 hotspots and 591 coldspots were obtained from the training datasets (Jiang et al., 2007a). In this dataset, there were 478 positive samples and 572 negative samples after removing redundancy using CD-HIT tool (Fu et al., 2012).

2.1.2. σ^{70} promoters

Promoters are regions in the DNA where RNA polymerase binds itself initiating the transcription process. The RNA polymerase combines itself with different σ factors, which are differentiated according to their nuclear weights. σ^{70} factors are primary housekeeping factors and hence have potential importance in gene transcription. The dataset that we have selected here for promoter sequence prediction is taken from (Lin et al., 2017). Originally the σ^{70} dataset was curated from the RegulonDB database (Santos-Zavaleta et al., 2019). In recent years, a large number of methods have been proposed to solve the promoter detection problem using this dataset (Lin et al., 2017; Liu et al., 2018; Rahman et al., 2019b,a). In this dataset, the promoter sequences are all DNA short sequences, and there are 741 positive and 1400 negative sequences.

Table 1
Summary of the different datasets used to test the performance of *SubFeat*.

Dataset	Sequence type	Positive instances	Negative instances	Total
Recombination hotspot	DNA	478	572	1050
σ^{70} promoters	DNA	741	1400	2141
RNA editing	RNA	125	119	244
DNA binding proteins	Protein	525	550	1075

2.1.3. RNA editing

Adenosine to inosine (A-to-I) editing is one of the most common and important RNA modifications (Peng et al., 2018) that changes the gene templates and thus affects the genetic variation in species. RNA–DNA difference (RDD) methods are generally employed to detect editing or modifications (Peng et al., 2012). Many machine learning-based methods are employed to approach the problem in recent years (Choyon et al., 2020; Ahmad and Shatabda, 2019; Chen et al., 2016). The dataset that we are using in this work was originally proposed in PAI (Chen et al., 2016). The proposed method was built based on St Laurent et al. (2013). They obtained a training dataset including 127 A-to-I editing sites containing sequences and 127 non-A-to-I editing sites containing sequences by sequencing the DNAs and RNAs of the wild-type D. melanogaster and RNAs of the ADAR-deficient *Drosophila melanogaster* by using a single molecular sequencing method. A benchmark dataset including 125 A-to-I editing sites containing sequences and 119 non-A-to-I editing sites containing sequences was obtained, after removing the redundant sample dataset were obtained by removing the redundant dataset. After preliminary trials, a length of the sequence in the benchmark dataset 51-nt was got, with that A. In the center, it can be edited to inosine, and all the sequence of the dataset is available at http://lin.uestc.edu.cn/server/PAI. They built an independent dataset to verify the power of the proposed method; Using CD-HIT (Fu et al., 2012), they obtained 300 A-to-I editing site containing sequences. by removing more than 75% sequence similarity and harvesting the A-to-I editing site containing sequences of *D. melanogaster* from Yu and his

colleagues' work (Chen et al., 2016). It contains 300 length RNA sequences with 125 positive and 119 negative sequences. These sequences are also 51-nt long and are available at http://lin.uestc.edu.cn/server/PAI.

2.1.4. DNA binding proteins

DNA binding proteins bind to specific regions of DNA and affect gene regulation. In this paper, we have used a very well-used benchmark dataset for DNA binding proteins with 525 positive and 550 negative samples. This dataset was originally proposed in Liu et al. (2014) and has been used extensively in the literature (Chowdhury et al., 2017; Zaman et al., 2017; Rahman et al., 2018; Liu et al., 2014; Wei et al., 2017).

2.2. Feature representation

After the data collection, an essential step in machine learning-based methods is to convert the problem instances to a vector representation. Generally, the feature vector is a collection of properties:

$$F = \{f_1, f_2, \ldots, f_n\} \tag{2}$$

Different feature representation techniques have been used in the literature that includes: structural information (Islam et al., 2018), evolution properties (Uddin et al., 2018; Shatabda et al., 2017). However, in recent works, it has been shown that sequence-based features through elementary and simple to generate are most effective if selected or designed properly (Muhammod et al., 2019; Rahman et al., 2018). Moreover, our main objective in this work was to provide a generic framework for all three types of sequences and to reduce the complexity in the feature generation step. That is the reason that we have selected to use sequence-based features only. However, the framework still supports other features based on derived or secondary properties and usable wherever necessary and useful.

For the sake of simplicity in the experiments, we have selected a similar group of features for all three types of sequences: Monomer composition, di-mer composition, trimer composition, 1-gapped di-mono composition, and 1-gapped mono-di compositions. However, based on the alphabet size, the number of features extracted is different. We have used PyFeat tool (Muhammod et al., 2019) for feature extraction. Considering no overlaps, these features are then divided into three groups. The details of the features are given in Tables 2 and 3 .

2.3. SubFeat algorithm

The pseudo-code of *SubFeat* algorithm is given in Algorithm 1. It follows the same procedure as described in Figs. 1–3 . However, given a set of instances in the training set, X and the labels associated with the y, the algorithm first extracts the feature set, F. From, F, next it populates a feature subspace set, $\mathbb{X}_s$. This set contains all the subspaces. Which is controlled by two parameters, n_p denoting the number of partitions in the feature space, and overlap is a Boolean indicating whether there will be overlaps among the subspaces or not. In practice, n_p and overlap could be hyper-parameters and needs to be trained based on a specific problem in concern. After that, iteratively, the hypothesis set, $\mathbb{H}$ and associated weights, $\mathbb{W}$ are learned based on the classifier type selected.

Table 2
Details of feature subspacing for protein dataset.

Feature subspace	Feature type	No. of features
F_1	MonoMer composition	20
	DiMer composition	400
	TriMer composition	8000
F_2	1-Gapped di-mono composition	8000
F_3	1-Gapped mono-di composition	8000

Table 3
Details of feature subspacing for DNA and RNA dataset.

Feature subspace	Feature type	No. of features
F_1	MonoMer composition	4
	DiMer composition	16
	TriMer composition	64
F_2	1-Gapped di-mono composition	64
F_3	1-Gapped mono-di composition	64

Algorithm 1. *SubFeat*$(X, y, n_p = 3, \text{overlap} = \text{false})$

$F = \text{extractFeatures}(X)$
Let $\mathbb{X}_s = \{\}$, set of features sub-spaces
Let $\mathbb{H} = \{\}$, set of learned hypothesis
Let $\mathbb{W} = \{\}$, set of weights of models
$X_S = \text{groupFeatures}(F, n_p, overlap)$
for *each* $X_i \in \mathbb{X}_s$ **do**
  $c_i = \text{selectClassifier}(mix = true)$
  $h_i = \text{learnClassifier}(X_i, y)$
  $w_i = \text{getWeight}\ (X_i, y, h_i)$
  $\mathbb{H} = \mathbb{H} \cup h_i$
  $\mathbb{W} = \mathbb{W} \cup w_i$
end
return $(\mathbb{W}, \mathbb{H})$

For prediction, the hypothesis set, $\mathbb{H}$ and weights set $\mathbb{W}$ are used to ensemble the predictions of the individual base classifiers in a weighted majority fashion. The parameter mix allows the mix of the models selected.

2.4. Performance evaluation

In this paper, we have divided the feature space 2–3 mers and 1–3 gaps based on categories, After that we select some feature of that category randomly (Tables 2 and 3). We have used 10-fold cross-validation for the sampling of the datasets. The dataset is divided into ten different balanced subsets retaining the balance ratio, and then in each iteration, one subset is used as a test, and the rest are taken as a train set. This process continued ten times. However, to tackle the randomness effect, ten runs were performed, and an average of them are reported only.

We have used several evaluation metrics: Accuracy (Acc), Precision, F1 Score, MCC, Sensitivity (Sn), Specificity (Spc), and Area under the curve (AUC). They are presented here in brief. Please note that in the following equations, TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative. True positive means positive instances that were correctly classified by the classifier. True negative means negative instances that were correctly classified by the classifier. Similarly, false positive and false negative means negative instances that are incorrectly classified as positive by the classifier and positive examples incorrectly classified as unfavorable by the classifier.

(3) **Accuracy (Acc)** gives a percentage result of correctly classified instances in between total number of instances.

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{3}$$

(4) **Sensitivity (Sn)** gives a percentage result of correctly classified positive instances in between total number of positive instances.

$$\text{Sn} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

(5) **Specificity (Spc)** gives a percentage result of correctly classified negative instances in between total number of negative instances:

$$\text{Spc} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{5}$$

(6) **Matthew's Correlation Coefficient (MCC)** returns value between +1 to − 1. The 0 represent a random classifier. The more the value is closer to +1, the better the classifier, similarly values towards − 1 represent bad classifier:

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FN} \times \text{FP})}{\sqrt{(\text{TP} + \text{FN}) \times (\text{TN} + \text{FP}) \times (\text{TP} + \text{FP}) \times (\text{TN} + \text{FN})}} \tag{6}$$

(7) **F1 score** is the weighted average of precision and Recall. F1-score works with both false positive and false negative. Especially in the term of an uneven class distribution, this metric is usually more useful than accuracy:

$$F1 - \text{Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{7}$$

Precision gives a result of correctly classified positive instances in between total number of positive instances:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

Recall is same as sensitivity and it is the ratio of correctly predicted true positive and false positive (all positive observations). It works on binary classification.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

(8) **Area under the receiver operating characteristic curve** (AUC) is a performance measurement for classification problems at various thresholds. AUC is the measure or degree of separability while ROC represents a probability curve.

3. Results and discussion

In this section, we present all the experimental results achieved in this study and relevant analysis. All the experiments were done in a Computing Machine provided by Bioinformatics Research Lab, United International University. The machine was equipped with 8-core processors, each core having an Intel Core Processor (i7-7700) with 3.6 GHz speed and 32 GB of memory. All experiments in this paper are implemented in Python programming language (Version 3.6) and using scikit-learn machine learning library (Pedregosa et al., 2011), and each of the experiments was run ten times, and the average of the results is reported. In all the tables, boldfaced values mean the best bargains.

3.1. Classification algorithms

In this section, we briefly describe the single-based classifiers and the ensemble. In this section, we briefly describe the single-based classifiers and the ensembles that were used for the experiments and for performance comparisons. Four single classifiers were used: support vector machines (SVM), Naive Bayes (NB), Decision Tree (DT), and logistic regression (LR). Support vector machine (SVM) (Cortes and Vapnik, 1995) selects vectors that can represent the decision boundary best to separate the different classes. In our experiments, we have used a linear kernel-based SVM. Logistic regression (LR) (Hosmer et al., 2013) divides the sample space using linear hyper-planes. We use L2 regularization and regularization parameters set to 1.0 for the experiments with iterations to learn the parameters to 100. Decision Tree (Ruggieri, 2002) is based on selecting features based on a measurement that can discriminate the instances best according to criteria. We used *gini index* as the selection criteria, and min samples to split were set 2. Gaussian Naive Bayesian (NB) (Jiang et al., 2007b) is supervised learning based on probabilities of the features given the class labels and their likelihoods.

In addition to these single classifiers, we have used three ensemble algorithms for experiments: AdaBoost, Random Forest, and Ensemble Voting. Each of these algorithms is state-of-the-art ensemble methods that are used in the bioinformatics domain and as well as in other areas (Rayhan et al., 2017; Li et al., 2018).

3.2. Experimental results

We present the results obtained by running experiments on four of the datasets. Tables 4–7 shows the result of using single classification, feature sub spacing ensemble classification, and different ensemble classifier like the random forest, AdaBoost, and ensemble voting algorithms on Recombination Hotspots, σ^{70} promoters, RNA editing, and DNA binding proteins problem respectively.

3.2.1. Recombination hotspot prediction

For the Recombination hotspot prediction dataset, the results are presented in Table 4. The first part of the table shows that among the single classifiers, Logistic regression performs significantly close. Since SVM and LR both are using linear decision boundaries, their performance very close to each other. However, when we turn to ensembles, we could notice that the Random Forest algorithm performs significantly better than other methods. In the lower part of the table, we present the results obtained by *SubFeat* using different combinations of single base classifiers. Note that, for this paper, we have used only three base classifiers. The performance of all decision tree combinations somewhat lacks compared to others. Among all these combinations, it appears that Naive Bayes and SVM combinations are working best. Here we can conclude that the mix of the base classifiers is not working well compared to the variety of using the same base classifiers. Also, note that these results by a right margin better than the results obtained by the ensemble methods.

Fig. 4 shows the area under the receiver operating characteristic curves analysis for the recombination hotspot dataset. In this figure, we also put the standard deviations among all the runs. We could notice that the proposed method shows higher performance, and over the different thresholds, its performance is superior to the other techniques, single or ensemble. The strong understanding of the proposed method, *SubFeat* in terms of AUC, provides evidence on the method's robustness.

3.2.2. σ70 promoters prediction

Table 5 presents the results of our experiments on the σ^{70} promoters prediction problem. Here too, we have presented the results in three parts: single, ensembles and *SubFeat* and its variations. We note that logistic regression outperforms the other methods from the results obtained in the single classifier experiments. However, once again, the performance of SVM is very close to logistic regression, which is expected. In the ensemble part, the results are improved compared to the single classifier results. Here, we could notice that Random FOest outperforms the rest of the methods. Moving to the third part of the table, we find the products of the different combinations of the single classifiers within the *SubFeat* framework. Similar to the results on the recombination hotspot problem, here too, we notice that the mixed combination of the single classifiers is not working as compared to the ensemble created with the same type of classifier. The best performing combination was produced by the Naive Bayes algorithm. SVM and logistic regression followed closely. Decision Tree combinations performed poorly. Also, note that this dataset was the largest among the datasets considered for this work.

Table 4
Experimental result on recombination hotspot prediction dataset.

Algorithm	Precision	F1	Acc	MCC	Sn	Spc	AUC
Result of single classifier algorithms							
SVM	0.8794	0.8016	0.7826	0.5633	0.7950	**0.7673**	0.8667
NB	0.7923	0.5975	0.6525	0.3636	**0.8065**	0.5794	0.7969
LR	**0.8839**	**0.8034**	**0.7854**	**0.5687**	0.8018	0.7658	**0.8687**
DT	0.7070	0.7547	0.7339	0.4659	0.7574	0.7061	0.7321
Result of using different ensemble classifiers							
Random Forest	**0.8913**	**0.8322**	**0.8120**	**0.6225**	0.8098	**0.8150**	**0.8874**
Adaboost	0.8589	0.7982	0.7760	0.5492	0.7827	0.7672	0.8497
Ensemble Voting	0.8794	0.7754	0.7699	0.5471	**0.8244**	0.7186	0.8654
Result of feature subspacing ensemble classification							
SVM+SVM+SVM	**0.9724**	0.8760	0.8464	0.7158	0.7835	**0.9865**	**0.9708**
NB+NB+NB	0.9681	**0.9078**	**0.8946**	**0.7911**	**0.8674**	0.9351	0.9647
LR+LR+LR	0.9697	0.8731	0.8420	0.7079	0.7787	0.9856	0.9706
DT+DT+DT	0.8562	0.8440	0.8297	0.6584	0.8420	0.8154	0.8771
SVM+NB+LR	0.9505	0.8871	0.8632	0.7420	0.8072	0.9739	0.9423
NB+LR+SVM	0.9498	0.8907	0.8676	0.7502	0.8113	0.9780	0.9441
LR+SVM+NB	0.9471	0.8925	0.8697	0.7548	0.8128	0.9813	0.9421
DT+SVM+DT	0.9194	0.8689	0.8483	0.6980	0.8227	0.8884	0.9079
SVM+DT+DT	0.9148	0.8684	0.8481	0.6976	0.8233	0.8871	0.9065
LR+LR+DT	0.9199	0.8852	0.8609	0.7366	0.8060	0.9689	0.9047
SVM+LR+DT	0.9182	0.8824	0.8582	0.7286	0.8070	0.9562	0.9032
SVM+NB+DT	0.9382	0.8916	0.8731	0.7513	0.8355	0.9354	0.9313

Table 5
Experimental result on σ70 promoters dataset.

Algorithm	Precision	F1	Acc	MCC	Sn	Spc	AUC
Result of single classifier algorithms							
SVM	0.8924	0.8238	0.7647	0.4721	0.8070	0.6742	0.8229
NB	0.8883	0.7936	0.7473	**0.4790**	**0.8501**	0.6093	0.818
LR	**0.8978**	**0.8262**	**0.7655**	0.4696	0.8013	**0.6835**	**0.8286**
DT	0.7377	0.7604	0.6881	0.3142	0.7638	0.5490	0.6574
Result of using different ensemble classifiers							
Random Forest	**0.9024**	**0.8331**	**0.7735**	0.4862	0.8036	**0.7020**	**0.8368**
Adaboost	0.8848	0.8106	0.7490	0.4399	0.7997	0.6452	0.8084
Ensemble Voting	0.8967	0.8188	0.7652	**0.4865**	**0.8255**	0.6563	0.8243
Result of feature subspacing ensemble classification							
SVM+SVM+SVM	0.9589	**0.8860**	0.8098	0.5664	0.8007	**0.8408**	**0.9232**
NB+NB+NB	0.9513	0.8556	**0.8203**	**0.6255**	**0.9008**	0.7038	0.9084
LR+LR+LR	**0.9598**	0.8552	0.7886	0.5170	0.7745	0.8470	0.9222
DT+DT+DT	0.8261	0.8227	0.7605	0.4577	0.7975	0.6758	0.7786
SVM+NB+LR	0.9442	0.8680	0.8175	0.5853	0.8233	0.8020	0.8969
NB+LR+SVM	0.9446	0.8663	0.8153	0.5796	0.8225	0.7962	0.8964
LR+SVM+NB	0.9443	0.8670	0.8166	0.5836	0.8240	0.7970	0.8964
DT+SVM+DT	0.9007	0.8406	0.7791	0.4935	0.7958	0.7336	0.8275
SVM+DT+DT	0.9021	0.8447	0.7857	0.5101	0.8023	0.7413	0.8320
LR+LR+DT	0.9178	0.8508	0.7862	0.5079	0.7829	0.7980	0.8414
SVM+LR+DT	0.9222	0.8563	0.7960	0.5326	0.7939	0.8031	0.8545
SVM+NB+DT	0.9297	0.8602	0.8119	0.5765	0.8367	0.7566	0.8736

The receiver operating characteristic analysis on the σ70 promoters prediction dataset are presented using a curve of false-positive rate against true positive rate and shown in Fig. 5. *SubFeat* method here outperforms the other techniques with an adequate margin again. Note that the threshold changes on the *x*-axis of the curve do not change the accurate favorable rates. For a balanced dataset chosen for the purpose, this is a strong indication of the superior performance of *SubFeat* over the other methods compared in this work.

3.2.3. A-to-I RNA editing site prediction

We present the experimental results on the A-to-I RNA editing sites prediction problem in Table 7. Note that this is a relatively smaller dataset compared to the other datasets. Here the performance of the single classifiers shown in the first part of the table is dominated by the logistic regression classifier in terms of all the performance metrics. Here, among the ensemble methods ensemble voting method performs significantly better compared to Random Forest or AdaBoost algorithms. However, *SubFeat* once again outperforms all these methods in terms of performance. This is clearly shown in the values reported in the lower part of the table. Here, we see that *SubFeat* follows the same trend as the previous datasets, that the ensemble is working better when the same classifier is chosen as the base classifier. However, Naive Bayes is performing slightly better, and SVM and logistic regression follow closely.

The ROC analysis for this dataset is shown in Fig. 6. Note that, for this dataset though *SubFeat* is still superior in performance in terms of AUC values, the difference is not that high as compared to the other datasets. Here, single classifier is working better compared to other datasets.

3.2.4. DNA binding proteins prediction

Experimental results on the DNA binding proteins prediction problem are reported in Table 7. We could note similar trends for this dataset as well. Logistic regression performs best in the single classifier group. Identical to that performance combination of logistic regression classifier used in the *SubFeat* is best among all the classification algorithms. The performance of this combination is slightly weaker in terms of AUC compared to the all SVM combination. This is due to the better precision values obtained by the SVM combination, which is also reflected in the specificity values reported in the table. The ROC analysis is shown in

Table 6
Experimental result on RNA editing dataset.

Algorithm	Precision	F1	Acc	MCC	Sn	Spc	AUC
Result of single classifier algorithms							
SVM	0.8788	0.7809	0.7918	0.5894	0.8019	0.7832	0.860
NB	0.8263	0.7359	0.7546	0.5165	0.7706	0.7407	0.7990
LR	**0.9021**	**0.8041**	**0.8128**	**0.6342**	**0.8182**	**0.8088**	**0.8823**
DT	0.6627	0.7087	0.7224	0.4535	0.7187	0.7256	0.7219
Result of using different ensemble classifiers							
Random Forest	0.8801	0.7379	0.7765	0.5724	**0.8500**	0.7315	0.8483
Adaboost	0.8153	0.7217	0.7409	0.4901	0.7476	0.7357	0.7887
Ensemble Voting	**0.9009**	**0.7779**	**0.7965**	**0.6048**	0.8184	**0.7794**	**0.8775**
Result of feature subspacing ensemble classification							
SVM+SVM+SVM	**0.9315**	0.8007	0.8310	0.6833	**0.9225**	0.7764	0.9137
NB+NB+NB	0.9155	**0.8386**	**0.8500**	**0.7065**	0.8694	**0.8339**	0.9059
LR+LR+LR	0.9302	0.8048	0.8276	0.6680	0.8860	0.7882	**0.9144**
DT+DT+DT	0.8251	0.8070	0.8106	0.6280	0.7993	0.8223	0.8619
SVM+NB+LR	0.8932	0.8070	0.8283	0.6692	0.8813	0.7904	0.88024
NB+LR+SVM	0.9012	0.7932	0.8219	0.6598	0.8892	0.7780	0.8896
LR+SVM+NB	0.9002	0.8060	0.8288	0.6704	0.8860	0.7890	0.8831
DT+SVM+DT	0.8993	0.8263	0.8382	0.6843	0.8647	0.8176	0.8876
SVM+DT+DT	0.8900	0.8106	0.8243	0.6553	0.8448	0.8083	0.8796
LR+LR+DT	0.8974	0.8116	0.8293	0.6686	0.8677	0.8011	0.8840
SVM+LR+DT	0.8779	0.7837	0.8097	0.6351	0.8659	0.7716	0.8584
SVM+NB+DT	0.8846	0.7999	0.8179	0.6659	0.8533	0.7918	0.8686

Table 7
Experimental result on DNA binding proteins dataset.

Algorithm	Precision	F1	Acc	MCC	Sn	Spc	AUC
Result of single classifier algorithms							
SVM	0.7925	0.6986	0.7108	0.4279	0.7472	0.6812	0.7849
NB	0.5512	0.6643	0.5754	0.1623	0.5577	0.6297	0.5708
LR	**0.8129**	**0.7303**	**0.7333**	**0.4696**	**0.7555**	**0.7130**	**0.7995**
DT	0.5864	0.6273	0.6189	0.2387	0.6272	0.6103	0.6187
Result of using different ensemble classifiers							
Random Forest	**0.7821**	0.7072	**0.7000**	**0.4009**	**0.7058**	0.6940	**0.7769**
Adaboost	0.7145	0.6760	0.6673	0.3358	0.6734	0.6611	0.7190
Ensemble Voting	0.7768	**0.7181**	0.6922	0.3879	0.6753	**0.7160**	0.7583
Result of feature subspacing ensemble classification							
SVM+SVM+SVM	**0.9051**	0.7741	0.7227	0.4833	0.6641	**0.8697**	**0.9004**
NB+NB+NB	0.6075	0.6908	0.5990	0.2256	0.5704	0.7042	0.6440
LR+LR+LR	0.8788	**0.8128**	**0.7903**	**0.5905**	**0.7488**	0.8542	0.8822
DT+DT+DT	0.6617	0.6694	0.6634	0.3276	0.6728	0.6538	0.6987
SVM+NB+LR	0.7923	0.7620	0.7105	0.4524	0.6578	0.8363	0.7618
NB+LR+SVM	0.7914	0.7602	0.7071	0.4465	0.6543	0.8357	0.7644
LR+SVM+NB	0.7900	0.7568	0.7055	0.4406	0.6550	0.8234	0.7615
DT+SVM+DT	0.7791	0.7288	0.7041	0.4117	0.6861	0.7291	0.7647
SVM+DT+DT	0.7810	0.7365	0.7120	0.4277	0.6918	0.7403	0.7714
LR+LR+DT	0.7914	0.7770	0.7538	0.5135	0.7241	0.7976	0.7744
SVM+LR+DT	0.7934	0.7715	0.7427	0.4944	0.7071	0.7993	0.7734
SVM+NB+DT	0.7704	0.7258	0.6735	0.3639	0.6362	0.7525	0.7477

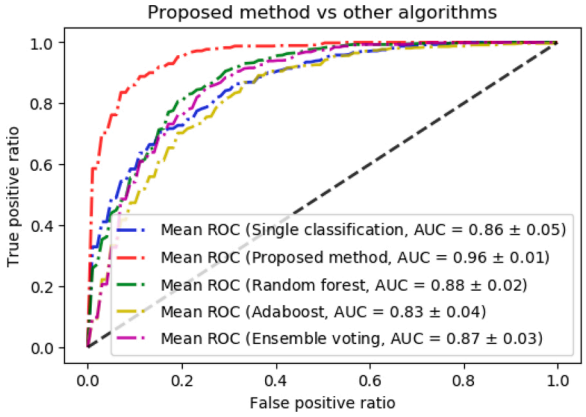

Fig. 4. ROC analysis for recombination hotspot problem dataset.

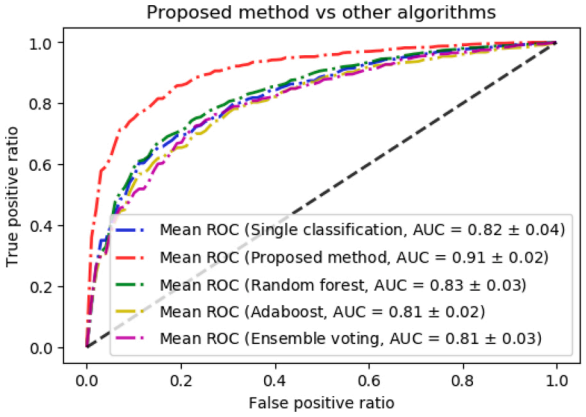

Fig. 5. ROC analysis for $\sigma 70$ promoters problem dataset.

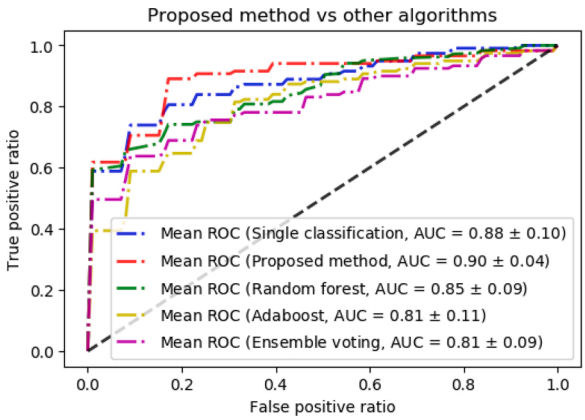

Fig. 6. ROC analysis for RNA editing prediction problem dataset.

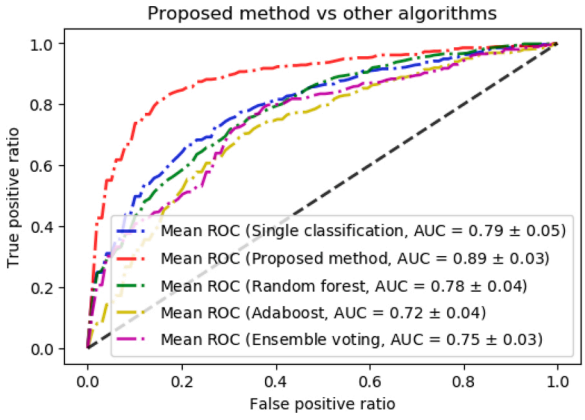

Fig. 7. ROC analysis for DNA binding proteins prediction problem dataset.

Fig. 7 in more details. The plot shows the superior performance of *SubFeat* over all other methods.

4. Comparison with other packages

The *SubFeat* tool works with different feature selection techniques that randomly selects from each category, which the user may customize for his experiments. We strongly believe if the user selects the particular feature in random tries, the result will increase. We have justified our hypothesis by experimentation with the four well-established datasets. The comparison of our using datasets and technical comparison with other software packages PyFeat (Muhammod et al., 2019), BioSeq-Analysis (Liu, 2019), iLearn (Chen et al., 2020) and iFeature (Chen et al., 2018) are given in Tables 8–11.

Table 8
Technical comparison on different packages.

Functionality	*SubFeat*	PyFeat	BioSeq-Analysis	iLearn	iFeature
Ensemble	Yes	No	No	Yes	Yes
Overlap	Yes	No	No	No	No
Overall functionality	Limited	Vast	Vast	Vast	Vast
Hyper-parameter Tuning	Limited	Limited	Vast	Vast	Vast
Feature generation ability	Limited	Vast	Vast	Vast	Vast
Cluster (machine learning)	No	No	No	Yes	Yes
Prediction (machine learning)	Yes	No	Yes	Yes	Yes
Online predictor	No	No	Yes	Yes	Yes

Table 9
Result comparison of RNA dataset on different packages.

Result	*SubFeat*	PyFeat	BioSeq-Analysis	iLearn	iFeature
Acc	83.23	76.23	76.73	77.13	80.00
AUC	0.9286	0.859	0.8297	0.8527	0.9027
MCC	0.6847	0.536	0.535	0.5672	0.6326

Table 10
Result comparison of DNA-binding/protein dataset on different packages.

Result	*SubFeat*	PyFeat	BioSeq-Analysis	iLearn	iFeature
Acc	82.52	78.28	74.73	76.57	79.74
AUC	0.9351	0.833	0.8097	0.8173	0.8927
MCC	0.6510	0.5090	0.506	0.5275	0.6526

Table 11
Result comparison of $\sigma 70$ promoters dataset on different packages.

Result	*SubFeat*	PyFeat	BioSeq-Analysis	iLearn	iFeature
Acc	80.98	67.91	76.37	75.97	79.12
AUC	0.9232	0.7438	0.8297	0.8173	0.8527
MCC	0.5664	0.3592	0.4726	0.5275	0.5526

5. Discussion

In this paper, we have proposed an ensemble method where the full feature space was divided into subspaces. From the results, we can conclude that the subspace method provides better prediction results than the single classifiers and the best ensemble algorithms like AdaBoost, Random Forest, etc. We have tested the algorithm's performance on a full space feature representation for protein, DNA, and RNA sequences datasets. However, it is possible to improve our accuracy by using a different feature space and feature selection techniques. We have only tested our method on balanced binary classification biological datasets. We have tested using overlaps of the feature-spaces; however, the number of the subspace is still a parameter to be tested comprehensively. Therefore, in the future, we plan to work with imbalanced data, independent and large numbers of the dataset. The simplicity of these method help to increase the accuracy of biological sequence datasets.

As a method, *SubFeat* shows better performance in all metrics than single and ensemble classifiers as found in the results and analysis offered in the previous section. That establishes the claim of the hypothesis of using an ensemble and dividing the feature space into subspaces. However, another subtle observation could be made from the results that using a similar classifier as a base classifier is achieving better results compared to the mix of the classifiers. This study was limited to four datasets, and this remains still a question to be explored in detail if the mix parameter can also bring good results. We believe that might be utilized as well. Two of the variables or parameters of the *SubFeat* framework are less explored in this paper. They are n_p, the number of partitions set to 3 in all the experiments, and overlap which is kept false for all the experiments.

We believe the answer to the performance largely depends on the feature space or the feature representation. In this work, we have limited to use of only sequence-based features. In problems like DNA binding protein prediction, we have noticed application of structural and evolutional features has been used successfully (Chowdhury et al., 2017; Zaman et al., 2017). In the cases of DNA and RNA sequences as well, the researchers have used many other types of feature representation techniques. Note that the knowledge number of partitions for the feature space will obviously be enhanced by the selection of such methods, as previously we have seen group-based feature selection to be performing better in a wide range of problems (Adilina et al., 2019; Islam et al.,

2018). However, in those works, the idea of the ensemble method was not explored. We kept the experimental setting simpler and thus not extended the feature space. We believe using a larger and enhanced feature space will improve the results.

Another parameter is the overlapping of the feature spaces. Though we have not reported these four datasets, we have seen that the overlap parameters are not working well. We observed that sensitivity suffers if we accept overlap too much. Note that in a previous work (Rahman et al., 2019b), overlapping has been found useful for promoter prediction. The results presented in this paper are much superior compared to the ones reported in Rahman et al. (2019b). However, note that the objective of this paper is limited to show the effectiveness of the ensemble based on feature subspacing.

5.1. Python Package

We have made our method, *SubFeat* available as a Python-based package. It is freely available for use from https://github.com/fazlulhaquejony/SubFeat. The package includes all the parameters that we have discussed and provided as an option for the method. A simple-to-follow user guide is also provided on how to install and use the package (e.g., command-line, experiments). We firmly believe that further exploration is possible for this package, and it will be useful for computational biologists working in the relevant fields.

6. Conclusion

In this paper, we have proposed an ensemble method where the full feature space was divided into subspaces. From the results, we can conclude that the subspace method provides better prediction results compared to both the single classifiers and the best ensemble algorithms like AdaBoost, Random Forest, etc. We have tested the performance of the algorithm on a full space feature representation for protein, DNA, and RNA sequences datasets. We have also tested our result by random selection and random overlapped selection. We also compared our proposed method. However, it is possible to improve our accuracy by using a different feature space and feature selection techniques. We have only tested our method on balanced binary classification biological datasets. We have tested using overlaps of the feature-spaces; however, the number of the subspace is still a parameter to be tested comprehensively. Therefore, in the future, we plan to work with imbalanced data, independent and large numbers of datasets. The simplicity of these methods helps to increase the accuracy of biological sequence datasets.

Authors' contribution

H.M.Fazlul Haque, Muhammod Rafsanjani and Fariha Arifin: methodology, software, formal analysis, data curation, writing – original draft. Sheikh Adilina: investigation, data curation, visualization. Swakkhar Shatabda: conceptualization, validation, writing – original draft, writing – review & editing, supervision, project administration.

Conflict of interest

None declared.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.compbiolchem.2021.107489.

References

Adilina, S., Farid, D.M., Shatabda, S., 2019. Effective dna binding protein prediction by using key features via Chou's general pseaac. J. Theoret. Biol. 460, 64–78.

Ahmad, A., Shatabda, S., 2019. Epai-nc: enhanced prediction of adenosine to inosine rna editing sites using nucleotide compositions. Anal. Biochem. 569, 16–21.

Ahmad, M.W., Arafat, M.E., Taherzadeh, G., Sharma, A., Dipta, S.R., Dehzangi, A., Shatabda, S., 2020. Mal-light: enhancing lysine malonylation sites prediction problem using evolutionary-based features. IEEE Access 8, 77888–77902.

Al Maruf, M.A., Shatabda, S., 2019. irspot-sf: prediction of recombination hotspots by incorporating sequence based features into Chou's pseudo components. Genomics 111 (4), 966–972.

Amin, R., Rahman, C.R., Ahmed, S., Sifat, M., Rahman, H., Liton, M.N.K., Rahman, M., Khan, M., Hossain, Z., Shatabda, S., et al., 2019. ipromoter-bncnn: a novel branched cnn based predictor for identifying and classifying sigma promoters. Bioinformatics.

Baudat, F., Buard, J., Grey, C., Fledel-Alon, A., Ober, C., Przeworski, M., Coop, G., De Massy, B., 2010. Prdm9 is a major determinant of meiotic recombination hotspots in humans and mice. Science 327 (5967), 836–840.

Bell, C.G., Lowe, R., Adams, P.D., Baccarelli, A.A., Beck, S., Bell, J.T., Christensen, B.C., Gladyshev, V.N., Heijmans, B.T., Horvath, S., et al., 2019. Dna methylation aging clocks: challenges and recommendations. Genome Biol. 20 (1), 249.

Chen, W., Feng, P., Ding, H., Lin, H., 2016. Pai: predicting adenosine to inosine editing sites by using pseudo nucleotide compositions. Sci. Rep. 6 (1), 1–7.

Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T.T., Wang, Y., Webb, G.I., Smith, A.I., Daly, R.J., Chou, K.-C., et al., 2018. ifeature: a python package and web server for features extraction and selection from protein and peptide sequences. Bioinformatics 34 (14), 2499–2502.

Chen, Z., Zhao, P., Li, F., Marquez-Lago, T.T., Leier, A., Revote, J., Zhu, Y., Powell, D.R., Akutsu, T., Webb, G.I., et al., 2020. ilearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of dna, rna and protein sequence data. Brief. Bioinformatics 21 (3), 1047–1057.

Chowdhury, S.Y., Shatabda, S., Dehzangi, A., 2017. idnaprot-es: identification of dna-binding proteins using evolutionary and structural features. Sci. Rep. 7 (1), 1–14.

Choyon, A., Rahman, A., Hasanuzzaman, M., Farid, D.M., Shatabda, S., 2020. Presa2i: incremental decision trees for prediction of adenosine to inosine rna editing sites. F1000Research 9 (262), 262.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297.

Fu, L., Niu, B., Zhu, Z., Wu, S., Li, W., 2012. Cd-hit: accelerated for clustering the next-generation sequencing data. Bioinformatics 28 (23), 3150–3152.

Gabernet, G., Gautschi, D., Müller, A.T., Neuhaus, C.S., Armbrecht, L., Dittrich, P.S., Hiss, J.A., Schneider, G., 2019. In silico design and optimization of selective membranolytic anticancer peptides. Sci. Rep. 9 (1), 1–11.

Hosmer Jr., D.W., Lemeshow, S., Sturdivant, R.X., 2013. Applied Logistic Regression, Vol.398. John Wiley & Sons.

Islam, M.M., Saha, S., Rahman, M.M., Shatabda, S., Farid, D.M., Dehzangi, A., 2018. iprotgly-ss: identifying protein glycation sites using sequence and structure based features. Proteins Struct. Funct. Bioinformatics 86 (7), 777–789.

Jani, M.R., Mozlish, M.T.K., Ahmed, S., Tahniat, N.S., Farid, D.M., Shatabda, S., 2018. irecspot-ef: effective sequence based features for recombination hotspot prediction. Comput. Biol. Med. 103, 17–23.

Jiang, P., Wu, H., Wei, J., Sang, F., Sun, X., Lu, Z., 2007a. Rf-dymhc: detecting the yeast meiotic recombination hotspots and coldspots by random forest model using gapped dinucleotide composition features. Nucleic Acids Res. 35 (suppl_2), W47–W51.

Jiang, L., Wang, D., Cai, Z., Yan, X., 2007b. Survey of improving naive bayes for classification. International Conference on Advanced Data Mining and Applications 134–145.

Kaushik, A.C., Mehmood, A., Dai, X., Wei, D.-Q., 2020. A comparative chemogenic analysis for predicting drug-target pair via machine learning approaches. Sci. Rep. 10 (1), 1–11.

Li, J., Huang, Y., Yang, X., Zhou, Y., Zhou, Y., 2018. Rnam5cfinder: a web-server for predicting rna 5-methylcytosine (m5c) sites based on random forest. Sci. Rep. 8 (1), 1–5.

Lin, H., Liang, Z.-Y., Tang, H., Chen, W., 2017. Identifying sigma70 promoters with novel pseudo nucleotide composition. IEEE/ACM Trans. Comput. Biol. Bioinformatics.

Liu, B., Xu, J., Lan, X., Xu, R., Zhou, J., Wang, X., Chou, K.-C., 2014. idna-prot| dis: identifying dna-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition. PLOS ONE 9 (9), e106691.

Liu, B., Wang, S., Long, R., Chou, K.-C., 2017. irspot-el: identify recombination spots with an ensemble learning approach. Bioinformatics 33 (1), 35–41.

Liu, B., Yang, F., Huang, D.-S., Chou, K.-C., 2018. ipromoter-2l: a two-layer predictor for identifying promoters and their types by multi-window-based pseknc. Bioinformatics 34 (1), 33–40.

Liu, B., 2019. Bioseq-analysis: a platform for dna, rna and protein sequence analysis based on machine learning approaches. Brief. Bioinformatics 20 (4), 1280–1294.

Luo, X., Chi, W., Deng, M., 2019. Deepprune: learning efficient and interpretable convolutional networks through weight pruning for predicting dna-protein binding. Front. Genet. 10, 1145.

Muhammod, R., Ahmed, S., Md Farid, D., Shatabda, S., Sharma, A., Dehzangi, A., 2019. Pyfeat: a python-based effective feature generation tool for dna, rna and protein sequences. Bioinformatics 35 (19), 3831–3833.

Namuduri, S., Narayanan, B.N., Karbaschi, M., Cooke, M., Bhansali, S., 2019. Automated quantification of dna damage via deep transfer learning based analysis of comet assay images. Applications of Machine Learning, Vol. 11139. International Society for Optics and Photonics, p. 111390Y.

Ning, Q., Ma, Z., Zhao, X., 2019. dforml (knn)-pseaac: detecting formylation sites from protein sequences using k-nearest neighbor algorithm via Chou's 5-step rule and pseudo components. J. Theoret. Biol. 470, 43–49.

Ntranos, V., Yi, L., Melsted, P., Pachter, L., 2019. A discriminative learning approach to differential expression analysis for single-cell rna-seq. Nat. Methods 16 (2), 163–166.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12 (Oct), 2825–2830.

Peng, Z., Cheng, Y., Tan, B.C.-M., Kang, L., Tian, Z., Zhu, Y., Zhang, W., Liang, Y., Hu, X., Tan, X., et al., 2012. Comprehensive analysis of rna-seq data reveals extensive rna editing in a human transcriptome. Nat. Biotechnol. 30 (3), 253–260.

Peng, X., Xu, X., Wang, Y., Hawke, D.H., Yu, S., Han, L., Zhou, Z., Mojumdar, K., Jeong, K.J., Labrie, M., et al., 2018. A-to-i rna editing contributes to proteomic diversity in cancer. Cancer Cell 33 (5), 817–828.

Rahman, M.S., Shatabda, S., Saha, S., Kaykobad, M., Rahman, M.S., 2018. Dpp-pseaac: a dna-binding protein prediction model using chou's general pseaac. J. Theoret. Biol. 452, 22–34.

Rahman, M.S., Aktar, U., Jani, M.R., Shatabda, S., 2019a. ipro70-fmwin: identifying sigma70 promoters using multiple windowing and minimal features. Mol. Genet. Genomics 294 (1), 69–84.

Rahman, M.S., Aktar, U., Jani, M.R., Shatabda, S., 2019b. ipromoter-fsen: identification of bacterial σ70 promoter sequences using feature subspace based ensemble classifier. Genomics 111 (5), 1160–1166.

Rashid, M.M., Shatabda, S., Hasan, M., Kurata, H., et al., 2020. Recent development of machine learning methods in microbial phosphorylation sites. Curr. Genomics 21 (3), 194–203.

Rayhan, F., Ahmed, S., Shatabda, S., Farid, D.M., Mousavian, Z., Dehzangi, A., Rahman, M.S., 2017. idti-esboost: identification of drug target interaction using evolutionary and structural features with boosting. Sci. Rep. 7 (1), 1–18.

Rimmer, A., Phan, H., Mathieson, I., Iqbal, Z., Twigg, S.R., Wilkie, A.O., McVean, G., Lunter, G., 2014. Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. Nat. Genet. 46 (8), 912–918.

Ruggieri, S., 2002. Efficient c4. 5 [classification algorithm]. IEEE Trans. Knowl. Data Eng. 14 (2), 438–444.

Santos-Zavaleta, A., Salgado, H., Gama-Castro, S., Sánchez-Pérez, M., Gómez-Romero, L., Ledezma-Tejeida, D., García-Sotelo, J.S., Alquicira-Hernández, K., Muñiz-Rascado, L. J., Peña-Loredo, P., et al., 2019. Regulondb v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in *E. coli* k-12. Nucleic Acids Res. 47 (D1), D212–D220.

Shatabda, S., Saha, S., Sharma, A., Dehzangi, A., 2017. iphloc-es: identification of bacteriophage protein locations using evolutionary and structural features. J. Theoret. Biol. 435, 229–237.

Singh, J., Hanson, J., Paliwal, K., Zhou, Y., 2019. Rna secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. Nat. Commun. 10 (1), 1–13.

St Laurent, G., Tackett, M.R., Nechkin, S., Shtokalo, D., Antonets, D., Savva, Y.A., Maloney, R., Kapranov, P., Lawrence, C.E., Reenan, R.A., 2013. Genome-wide analysis of a-to-i rna editing by single-molecule sequencing in drosophila. Nat. Struct. Mol. Biol. 20 (11), 1333.

Taherzadeh, G., Yang, Y., Zhang, T., Liew, A.W.-C., Zhou, Y., 2016. Sequence-based prediction of protein-peptide binding sites using support vector machine. J. Comput. Chem. 37 (13), 1223–1229.

Turan, M.K., Sehirli, E., 2017. A novel method to identify and grade dna damage on comet images. Comput. Methods Programs Biomed. 147, 19–27.

Uddin, M.R., Sharma, A., Farid, D.M., Rahman, M.M., Dehzangi, A., Shatabda, S., 2018. Evostruct-sub: an accurate gram-positive protein subcellular localization predictor using evolutionary and structural features. J. Theoret. Biol. 443, 138–146.

Wei, L., Tang, J., Zou, Q., 2017. Local-dpp: an improved dna-binding protein prediction method by exploring local evolutionary information. Inform. Sci. 384, 135–144.

Xu, H., Jia, P., Zhao, Z., 2020. Deep4mc: systematic assessment and computational prediction for dna n4-methylcytosine sites by deep learning. Brief. Bioinformatics.

Zaman, R., Chowdhury, S.Y., Rashid, M.A., Sharma, A., Dehzangi, A., Shatabda, S., 2017. Hmmbinder: DNA-binding protein prediction using hmm profile based features. BioMed Res. Int. 2017.

Zhou, X., Chai, H., Zhao, H., Luo, C.-H., Yang, Y., 2020. Imputing missing rna-sequencing data from dna methylation by using a transfer learning-based neural network. GigaScience 9 (7), giaa076.