METHOD ARTICLE
PRESa2i: incremental decision trees for prediction of Adenosine to Inosine RNA editing sites [version 1; peer review: awaiting peer review]

Alif Choyon, Ashiqur Rahman, Md. Hasanuzzaman, Dewan Md Farid, Swakkhar Shatabda
Department of Computer Science and Engineering, United International University, Dhaka, 1212, Bangladesh

Abstract
RNA editing is a very crucial cellular process affecting protein encoding and is sometimes correlated with the cause of fatal diseases, such as cancer. Thus knowledge about RNA editing sites in a RNA sequence is very important. Adenosine to Inosine (A-to-I) is the most common of the RNA editing events. In this paper, we present PRESa2i, a computation prediction tool for identification of A-to-I RNA editing sites in given RNA sequences. PRESa2i uses a simple, yet effective set of sequence based features generated from RNA sequences and a novel feature selection technique. It uses an incremental decision tree algorithm as the classification algorithm. On a standard benchmark dataset and independent set, it achieves 86.48% accuracy and 90.67% sensitivity and significantly outperforms state-of-the-art methods. We have also implemented a web application based on PRESa2i and made it available freely at: http://brl.uiu.ac.bd/presa2i/index.php. The materials for this paper are also available to use from: https://github.com/swakkhar/RNA-Editing/.

Keywords
RNA Editing Sites, Sequence Based Features, Feature Selection, Incremental Decision Tree

Corresponding author: Swakkhar Shatabda (sshatabda@gmail.com)
Author roles: Choyon A: Data Curation, Investigation, Methodology, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; Rahman A: Methodology, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; Hasanuzzaman M: Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; Farid DM: Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; Shatabda S: Conceptualization, Formal Analysis, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.
Grant information: The author(s) declared that no grants were involved in supporting this work.
Copyright: © 2020 Choyon A et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Choyon A, Rahman A, Hasanuzzaman M et al. PRESa2i: incremental decision trees for prediction of Adenosine to Inosine RNA editing sites [version 1; peer review: awaiting peer review] F1000Research 2020, 9:262 (https://doi.org/10.12688/f1000research.22823.1)

First published: 16 Apr 2020, 9:262 (https://doi.org/10.12688/f1000research.22823.1)
Introduction

RNA editing is a process by which insertion, deletion or substitution of base nucleotides happen within the RNA. Such editing changes occur within the cell meaning nucleotide sequences are altered. This may cause alteration in the peptide sequence, which could be entirely different from that encoded in the DNA. Adenosine to Inosine (A-to-I) is the most common of all RNA editing events. Apart from making genetic alterations and resulting in codon changes A-to-I is often responsible for several intracellular operations like alternative splicing. A-to-I is found to have its effect on RNA secondary structural changes. Moreover, it is found to be in correlation with cancer formation. Thus, research and identification of RNA editing is very important.

Experimental methods like RNA-Seq are effective means for identification of A-to-I sites. However, they are time consuming. Moreover, as known RNA-editing sites are available, computational methods for prediction of RNA editing sites are becoming more relevant in this era of knowledge and data discovery. Various computational methods are proposed in the literature for prediction of RNA editing sites, including Genetic Algorithms (GA), Support Vector Machines (SVM), Logistic Regression (LR) and Deep Learning. PREPACT was proposed as a computational tool for C-to-U and U-to-C RNA editing site prediction for plant species. Another method was presented by 10 for prediction of C-to-U editing sites using biochemical and evolutionary information.

Several machine learning based prediction methods are proposed in the literature of A-to-I RNA editing site identification in recent years. PAI was proposed by 11 that used pseudo nucleotide compositions as features with a SVM classification algorithm. They proposed and built two benchmark datasets derived from the Drosophila genome based on the work by 12. In a subsequent work, they proposed iRNA-AI using general form of pseudo nucleotide composition (PseKNC) with SVM and further improved the previous results on A-to-I datasets. A prediction method iRNA-3typeA13 was proposed to detect and identify three types of RNA-modifications at Adenosine sites. In a recent work, auto-encoders were used to develop PAI-SAE for prediction of A-to-I sites.

In this paper, we propose and present PRESa2i, a novel prediction method for A-to-I RNA Editing sites. In our proposed method, we have used the Hoeffding tree, an incremental decision tree for classification of samples. We have used sequence based features extracted from RNA samples collected as positive and negative instances in the dataset. We have used a novel feature selection method to select only the top 179 features. On a standard benchmark dataset PRESa2i has achieved a significant accuracy of 86.48% and on the independent test set it has a significant sensitivity of 90.67%. We have also made our method freely available for use by researchers at: http://brl.uiu.ac.bd/presa2i/index.php.

Methods

PRESa2i is trained with a dataset that has sequence based features. A novel feature selection algorithm is applied to select the top features and incremental decision tree algorithms are learned. The model is built on a standard benchmark training set and the performance of the model is tested using an independent set. As suggested by 14 and followed by many researchers, we follow five steps for establishing any good tool for attribute prediction of biological entities: i) selection of benchmark datasets, ii) feature generation, iii) selection of appropriate algorithm, iv) evaluation methods, and v) establishment of a web server.

Datasets

The first step in any machine learning based supervised prediction task is to select a standard set of data. The RNA A-to-I editing site prediction problem is formulated as a binary classification problem, where Adenine in a RNA sequence or subsequence is labeled with two classes: A-to-I or positive samples and non A-to-I or negative samples. A dataset can be formally defined as the following:

$$ S = S^+ \cup S^- $$

Here, $S$ denotes the dataset and positive and negative datasets are denoted by $S^+$ and $S^-$ respectively. The datasets that we have used is first used in 11 based on the work in 12. The original dataset constructed in 12 contained 127 A-to-I editing sites with sequences and 127 non A-to-I editing sites with sequences. These sequences were obtained by sequencing wild-type and ADAR-deficient D. melanogaster DNA and RNA. After removing the redundant sequences, a benchmark dataset was constructed that contained 125 positive or A-to-I sites with sequences and 119 negative or non A-to-I sequence sites. Each of these RNA sequences in the dataset are 51 nucleotides long. From herein, we refer to this dataset as benchmark dataset.

An independent test set was constructed by further analyzing the sequences of D. melanogaster by 19. This set contained 51 nucleotide long 300 positive A-to-I sites with sequence. A summary of the datasets are presented in Table 1. Note that the benchmark dataset is balanced, while the independent dataset contains only positive sequences. Both datasets used are available here: https://github.com/swakkhar/RNA-Editing/.

| Dataset               | Positive Instances | Negative Instances | Sequence Size |
|-----------------------|--------------------|---------------------|---------------|
| Benchmark Dataset     | 125                | 119                 | 51            |
| Independent Test Set  | 300                | –                   | 51            |
RNA sequence representation

Any sample in the dataset is a sequence of RNAs. These are all 51 nucleotide long RNA strings. All of these strings are formed by taking symbols from the RNA alphabet, \( \Sigma = \{ A, C, G, U \} \). Formally a RNA sequence string \( R \in \Sigma \) can be formulated as below:

\[
R = N_1, N_2, \cdots, N_L
\]

(2)

Here, \( N_i \) is a nucleotide symbol and \( L = 51 \) is the length of the RNA sequence. We have extracted three groups of features from each sequence. Thus each sample in the dataset corresponds to a feature vector containing three groups of features, as below:

\[
F(R) = [F_1(R)F_2(R)F_3(R)]
\]

(3)

The three groups of features that were considered in the PRESa2i method are: \( k \)-mer compositions, gapped \( k \)-mer compositions and other statistical features. The first two types of features are widely used in the literature for solving other problems as well.

1. \( k \)-mer Compositions: \( k \)-mers are sequences of length \( k \) drawn from the alphabet \( \Sigma \). Compositions are calculated as the frequency of the different length \( k \)-mers normalized by sequence length. This is a widely used feature in the literature\(^{15,20}\). We have used \( k \)-mers with value of \( k = 1, 2, 3, 4 \). Thus the total number of features in this group was \( 4+16+64+256 = 340 \).

2. Gapped \( k \)-mers: We have used gapped \( k \)-mer compositions as features. These features were previously used in the literature\(^{11,21}\) and are extended from the ideas of gapped \( k \)-mer composition\(^1\) or gapped di-peptide composition\(^{22}\). They are normalized frequency of \( k \)-mers with gaps in between them. We have considered gaps, \( g = 1, 2, 3, \cdots, 10 \) in this paper and considered up to \( k \)-mers with \( k = 2, 3, 4 \). Here, the total number of features is \( 160+640+640+2560 = 4000 \).

3. Other statistical features: We have used other statistically derived features, ratio of start and end codons and distribution of bases, as features. In total, there were five features from this group. The start codon is AUG and end codons are UGA, UAA and UAG. We have used the ratio of these codons. Also there are four nucleotides or bases: A, U, G, C. We have converted the presence of these bases into a distribution and used that as a feature.

Hybrid feature selection

Total number of features generated using the feature extraction technique was 4345. The feature generation technique is simple since they are extracted directly from RNA sequences. After the features are selected, we have used a hybrid feature selection technique. Our feature hybrid feature selection method is a multi-step step method where the first step is a wrapper method followed by consecutive filter methods. In the first phase of feature selection we have used best first search (BFS). It is an incremental feature selection procedure that adds or deletes one feature at a time and finds the optimal set of features. After the first phase the selected features were then sent for a single pass classification done by a logistic function (LF). For each class labels, the features with positive weights were considered and others were filtered out in this phase. In the last step, we used ensemble of random decision trees and added most significant features found by the randomly generated features which were left in the previous step to get the final set of features.

Figure 1 shows the steps of the feature selection technique used by the PRESa2i method.

Incremental decision trees

We have used Hoeffding tree\(^{24}\), an incremental decision tree algorithm as the classification engine for PRESa2i. Though its more applicable to data streams, the performance of this classifier is nearly the same as non-incremental learning algorithms. This algorithm starts with a single leaf decision tree and in each iteration for all examples it uses the Hoeffding bound to split the node, recursively selecting one attribute each time based on evaluative functions. The leaves are decided using adaptive naive Bayes technique.

Performance evaluation

We have used four metrics for the binary classification problem. These are Accuracy, Sensitivity (\( S_n \)), Specificity (\( S_p \)) and Mathew’s Correlation Coefficient (MCC). These metrics were selected to conform with previous methods\(^{8,11}\).
For a binary classification method, let \( TP \) denote the number of true positives, \( TN \) denote the number of true negatives, \( FP \) denote the number of false positives and \( FN \) denote the number of false negatives. Now the four metrics are defined as follows:

\[
\text{Accuracy} = \frac{TN + TP}{TP + FN + TN + FP} \tag{4}
\]

\[
S_n = \frac{TP}{TP + FN} \tag{5}
\]

\[
S_p = \frac{TN}{TN + FP} \tag{6}
\]

\[
\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{7}
\]

Note that, the first three metrics have the values in the range \([0,1]\), where a value equal or near 0 means a bad classifier and a value near 1 means a good classifier. However, \( MCC \) have values in range \([-1,1]\). We have also used Receiver Operating Characteristic (ROC) curve to show the performances of different classifiers. It plots true positive rate against false positive rate at different thresholds of a probabilistic classification method.

\section*{Results and discussion}

In this section, we present the details of experimental results for our proposed method, PRESa2i. All the algorithms and programs were developed using Java 8 standard edition and Weka library\(^{27}\). All the experiments were done 10 times and only the average results are reported.

\section*{Comparison with other methods}

In this section, we provide the experimental results for comparisons with other state-of-the-art methods with our method PRESa2i. We have considered two previous state-of-the-art methods for comparison with our method: PAI proposed by 11 and PAI-SAE proposed by 8. We have used accuracy, sensitivity, specificity and \( MCC \) to compare the results. We have not run PAI and PAI-SE, rather as all these methods are tested on the same benchmark dataset, we have taken the results as being reported in the respective publications. Table 2 presents the results on the benchmark dataset. In this table, best values in each criteria are shown in bold fonts.

From the values reported in Table 2, we can see that PRESa2i achieves higher values in all the performance metrics compared with PAI and PAI-SAE. In case of accuracy our method is 6.89% and 4.51% more accurate than PAI and PAI-SAE, respectively. It also outperforms both of the methods in terms of MCC and sensitivity. In case of specificity its performance is significantly improved compared to other two methods.

To further assess the performance of our method we used an independent test set. Note that, while constructing the independent test set CD-HIT was used to remove sequences with higher similarity. We used the same cutoff as being used in PAI proposed by 11. In the case of the independent test set as there were no negative samples, and the only metric applicable was sensitivity. We report the sensitivity of PAI and PRESa2i in Table 3.

Note that, the sensitivity of our method on the benchmark dataset was not a result of overfit and hence 272 samples out of 300 samples were correctly predicted in the independent test set by PRESa2i compared to 247 correctly predicted by PAI. Thus, we can conclude that our method achieved superior performances in both benchmark dataset and independent test set over these current state-of-the-art methods.

\section*{Effects of feature selection}

We have proposed a multi-stage novel feature selection method. To show the effectiveness of our feature selection method, we have used accuracy, sensitivity and specificity as performance measures. We have compared the performance of the feature selection method with that of ‘Using All Features’ and with that of ‘Reduced Using BFS+LF’. Results are reported in Table 4. We have used six classifiers: Naive Bayes Classifier, AdaBoost Classifier, Hoeffding Tree, Random Forest Classifier, Support Vector Machines and Bagging.

The bold fonts show the best values of accuracy, sensitivity and specificity for each of the classifiers on the different set of features selected at different stages of feature selection. From this, we can note the effectiveness of the over-all feature selection procedure. Here, we can see that, using best first search and logistic function based feature selection in the first phase generates features that improves the performance of all the classifiers compared to no feature selection. On the other hand, when we further refine the features using decision tree based significance test, all the measures improve for four of the classifiers which is an indication of the effectiveness of the later stage of feature selection.

\section*{Classifier selection}

We can note that among the six classifiers used in our experiments as shown in Table 4, AdaBoost classifier and Hoeffding

### Table 2. Performance comparison of different methods - PRESa2i, PAI\(^{\text{11}}\) and PAI-SAE\(^{\text{8}}\) - on the benchmark dataset.

| Method name | Accuracy | MCC  | Sensitivity | Specificity |
|-------------|----------|------|-------------|-------------|
| PAI         | 79.59%   | 0.0600 | 0.8560     | 0.7311     |
| PAI-SAE     | 81.97%   | 0.6414 | 0.8720     | 0.7647     |
| PRESa2i     | 86.48%   | 0.7393 | 0.9520     | 0.7731     |

### Table 3. Performance comparison of different methods - PRESa2i, PAI\(^{\text{11}}\) - on the independent test set.

| Method Name | PAI\(^{\text{11}}\) | PRESa2i |
|-------------|-------------------|---------|
| Sensitivity | 82.33%            | 90.67%  |
Tree are the two best classifiers, followed by SVM. In the case of SVM though the accuracy is higher compared to these two algorithms, the sensitivity is not as good as these two. Note that these are the results on the benchmark dataset. We have also shown the receiver operating characteristic curve for each of these classifiers in Figure 2.

To make the selection of the final classification algorithm, we also tested them on the independent dataset. In Table 5, the results on the independent test set in terms of sensitivity is presented. We see that, the performance of Hoeffding tree does not overfit the data in benchmark dataset and hence the accuracy is enhanced from 86.47% to 90.67% from benchmark dataset to the independent test set. Thus we chose Hoeffding tree as our final classifier and save the model learned by this algorithm for the prediction tool.

Effect of negative test data
We have further investigated the effect of introducing negative instances in the independent test set. For this purpose, we have generated negative instances using a empirical distribution learned from the negative training dataset based on the distribution of the base nucleotides only. The generated negative instance are then added to the positive ones and the algorithms were again tested on this set. The results are reported in Table 6.

From the results reported in Table 6, we note that addition of the negative data in the independent test set does not affect on the effectiveness of the features and the feature selection method and the model trained. Note that, the introduced negative instances are given in the repository at: https://github.com/swakkhar/RNA-Editing/.

Please note that the results presented in Table 6 shows the effect of introducing negative examples in the dataset. Note that these negative datasets are taken randomly and are not verified at all. We note that though the accuracy and sensitivity is still satisfactory, Hoeffding tree and AdaBoost fails in terms of specificity. However, it also indicates that there is scope to improve the performance by adding more negative samples in the training set and also to improve the independent test set by adding real negative examples instead of taking random samples.

Web server
We have also implemented a web server, based on the best classification model learned by the incremental decision tree and final selected set of features on the benchmark dataset. Our website was developed using PHP language on the server side using Java and Weka library to provide the prediction from backend. Our method is readily available for use from: http://brl.uiu.ac.bd/presa2i/index.php.

Implementation
The web server has two major components: an web interface and a background server application. A small brief on the implementation is given below:

1. **Web Interface**: The web interface and its backend is implemented using PHP and bootstrap. The PHP script takes the input of the RNA sequence and then calls the prediction server which is a Java program and again receives the output from the prediction server and shows them in the web area.

2. **Prediction Server**: The prediction server is written in Java language. All the experiments are written in java language. The feature generation and selection part is implemented using Java. A program takes the input of the RNA sequence and generates the necessary features and then uses the Weka package for the classification of indentification engine. After the results are processed they are fed back to the Web interface where the results are shown. Interface between the Web Interface and the Java program is made using secured http connection.

Operation
The operation of the webserver is very simple. There are only one text field and two buttons. A screenshot of the developed

| Classifier       | Using All Features | Reduced Using BFS+LF | Final Selected Features |
|------------------|--------------------|----------------------|-------------------------|
|                  | Sn(%)  | Sp(%)  | Acc(%) | Sn(%)  | Sp(%)  | Acc(%) | Sn(%)  | Sp(%)  | Acc(%) |
| Naive Bayes      | 88.0   | 69.74  | 79.09  | 91.2   | 75.63  | 83.60  | 92.0   | 78.15  | 85.24  |
| AdaBoost         | 92.8   | 63.02  | 78.27  | 94.4   | 78.15  | 86.47  | 96.0   | 81.51  | 88.93  |
| Hoeffding Tree   | 90.4   | 68.90  | 79.91  | 92.8   | 76.47  | 84.83  | 95.2   | 77.31  | 86.47  |
| Random Forest    | 91.2   | 67.22  | 79.50  | 94.4   | 76.47  | 85.65  | 92.8   | 73.94  | 83.60  |
| SVM              | 82.4   | 77.31  | 79.91  | 92.8   | 88.23  | 90.57  | 93.6   | 89.91  | 91.80  |
| Bagging          | 88.0   | 64.70  | 76.63  | 93.6   | 71.42  | 82.78  | 88.0   | 70.58  | 79.50  |
Figure 2. Receiver operating characteristic curve for different classifiers on the benchmark dataset.

Table 5. Performance of different classifiers in terms of sensitivity on the independent test set.

| Algorithm Name        | Naive Bayes | SVM  | Hoeffding Tree | AdaBoost | Bagging | Random Forest |
|-----------------------|-------------|------|----------------|----------|---------|--------------|
| Sensitivity (%)       |             |      |                |          |         |              |
| Naive Bayes           | 86.67       | 64.67| 90.67          | 85.33    | 82.67   | 83.67        |

Table 6. Experiments on the independent dataset after introduction of negative instances.

| Algorithm       | $S_n$ (%) | $S_p$ (%) | Accuracy (%) |
|-----------------|-----------|-----------|--------------|
| Naive Bayes     | 86.66     | 73.33     | 80.00        |
| AdaBoost        | 85.33     | 51.33     | 68.33        |
| Hoeffding Tree  | 90.66     | 52.66     | 71.66        |
| Random Forest   | 83.66     | 74.33     | 79.00        |
| SVM             | 64.66     | 70.00     | 67.33        |
| Bagging         | 82.66     | 61.66     | 72.16        |

The web application is given in Figure 3. Here is a step by step procedure on how to use the web application:

1. First one has to put the input RNA sequences in the text area. If there is any mistake, pressing the ‘Clear’ button will clear the area.

2. After the input is given in the text area, the predict button should be pressed. Thus a request would be made to the online server.

3. The results for the query sequence will be shown just below the input pane. For the next entry, the process should be followed in the same manner starting from step 1.

Conclusion

In this article, we have proposed PRESa2i, a novel predictor for A-to-I RNA editing sites using sequence based features and incremental decision trees. Our method is based on a number of effective features selected using successive phases of best first search in incremental forward selection and ranking based on random decision trees and logistic regression. We have made all our datasets available online at: https://github.com/swakkhar/RNA-Editing/, and have also made our method available as an web application from: http://brl.uiu.ac.bd/presa2i/index.php. In the future, we would like to explore other RNA editing sites and develop layered or multi-class prediction system. We believe further enhancement of the
A dataset can improve the performance of the method to a great extent.

**Data availability**

Datasets for this article available from: https://github.com/swakkhar/RNA-Editing/

Archived datasets as at time of publication: https://zenodo.org/record/3727221

License: CC0

**Software availability**

PRESa2i web application: http://brl.uiu.ac.bd/presa2i/index.php

Source code available from: https://github.com/swakkhar/RNA-Editing

Archived source code as at time of publication: https://zenodo.org/record/3727221

---

**References**

1. Rueter SM, Dawson TR, Emeson RB: Regulation of alternative splicing by RNA editing. Nature. 1999; 399(6731): 75–80. PubMed Abstract | Publisher Full Text

2. Maas Stefan: Gene regulation through RNA editing. Discov Med. 2010; 10(54): 379–386. Published Abstract

3. Paz N, Levinon EY, Amariglio N, et al.: Altered adenosine-to-inosine RNA editing in human cancer. Genome Res. 2007; 17(11): 1586–1595. PubMed Abstract | Publisher Full Text | Free Full Text

4. Wang Z, Genstein M, Snyder M: RNA-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet. 2009; 10(1): 57. PubMed Abstract | Publisher Full Text | Free Full Text

5. Thompson J, Gopal S: Genetic algorithm learning as a robust approach to RNA editing site prediction. BMC Bioinformatics. 2006; 7(1): 145. PubMed Abstract | Publisher Full Text | Free Full Text

6. Chen W, Feng P, Yang H, et al.: iRNA-AI: identifying the adenosine to inosine editing sites in RNA sequences. Oncotarget. 2017; 8(3): 4208–4217. PubMed Abstract | Publisher Full Text | Free Full Text

7. Xiong H, Liu D, Li Q, et al.: RED-ML: a novel, effective RNA editing detection method based on machine learning. GigaScience. 2017; 6(5): 1–8. PubMed Abstract | Publisher Full Text | Free Full Text

8. Xiao X, Wang P, Xu Z, et al.: Pai-sae: Predicting adenosine to inosine editing sites based on hybrid features by using sparse autoencoder. In IOP Conference Series: Earth and Environmental Science. 2018; 170(5): 052018. IOP Publishing. Publisher Full Text

9. Lenz H, Knoop V: PREPACT 2.0: Predicting C-to-U and U-to-C RNA Editing in Organelle Genome Sequences with Multiple References and Curated RNA Editing Annotation. Bioinform Biol Insights. 2013; 7: 1–19. PubMed Abstract | Publisher Full Text | Free Full Text

10. Du P, Li Y: Prediction of C-to-U RNA editing sites in plant mitochondria using both biochemical and evolutionary information. J Theor Biol. 2008; 253(3): 579–586. PubMed Abstract | Publisher Full Text

11. Chen W, Feng P, Ding H, et al.: PAI: Predicting adenosine to inosine editing sites by using pseudo nucleotide compositions. Sci Rep. 2016; 6: 35123. PubMed Abstract | Publisher Full Text | Free Full Text

12. St Laurent G, Tackett Mr, Nechkin S, et al.: Genome-wide analysis of A-to-I RNA editing by single-molecule sequencing in Drosophila. Nat Struct Mol Biol. 2013;
13. Chen W, Feng P, Yang H, et al.: iRNA-3typeA: Identifying Three Types of Modification at RNA’s Adenosine Sites. Mol Ther Nucleic Acids. 2018; 11: 468-474. PubMed Abstract | Publisher Full Text

14. Chou KC: Some remarks on protein attribute prediction and pseudo amino acid composition. J Theor Biol. 2011; 273(1): 236–247. PubMed Abstract | Publisher Full Text | Free Full Text

15. Rahman S, Akbar U, Jani R, et al.: iPromoter-FSEn: Identification of bacterial σ70 promoter sequences using feature subspace based ensemble classifier. Genomics. 2019; 111(5): 1160–1166. PubMed Abstract | Publisher Full Text

16. Rayhan F, Ahmed S, Shatabda S, et al.: iDTI-ESBoost: Identification of Drug Target Interaction Using Evolutionary and Structural Features with Boosting. Sci Rep. 2017; 7(1): 17731. PubMed Abstract | Publisher Full Text | Free Full Text

17. Chowdhury SY, Shatabda S, Dehzangi A: iDNAProt-ES: Identification of DNA-binding Proteins Using Evolutionary and Structural Features. Sci Rep. 2017; 7(1): 14938. PubMed Abstract | Publisher Full Text | Free Full Text

18. Uddin MR, Sharma A, Farid DM, et al.: EvoStruct-Sub: An accurate Gram-positive protein subcellular localization predictor using evolutionary and structural features. J Theor Biol. 2018; 443: 138–146. PubMed Abstract | Publisher Full Text

19. Yu Y, Zhou H, Kong Y, et al.: The Landscape of A-to-I RNA Editome Is Shaped by Both Positive and Purifying Selection. PLoS Genet. 2016; 12(7): e1006191. PubMed Abstract | Publisher Full Text | Free Full Text

20. Shatabda S, Saha S, Sharma A, et al.: iPHLoc-ES: Identification of bacteriophage protein locations using evolutionary and structural features. J Theor Biol. 2017; 435: 229–237. PubMed Abstract | Publisher Full Text

21. Ali Mard MA, Shatabda S: iRSpot-SF: Prediction of recombination hotspots by incorporating sequence based features into Chou’s Pseudo. Genomics. 2019; 111(4): 966–972. PubMed Abstract | Publisher Full Text

22. Ghani M, Mohammad-Noori M, Beer MA: Robust k-mer frequency estimation using gapped k-mers. J Math Biol. 2014; 69(2): 469–500. PubMed Abstract | Publisher Full Text | Free Full Text

23. Chang J, Su EC, LO A, et al.: PSLDoc: Protein subcellular localization prediction based on gapped-dipeptides and probabilistic latent semantic analysis. Proteins. 2008; 72(2): 693–710. PubMed Abstract | Publisher Full Text

24. Hulten G, Spencer L, Domingos P: Mining time-changing data streams. In ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining. ACM Press. 2001; 97–106. Publisher Full Text

25. Hall M, Frank E, Holmes G, et al.: The weka data mining software: an update. ACM SIGKDD explorations newsletter. 2009; 11(1): 10–18. Publisher Full Text

26. Chen X, Tang H, Li WC, et al.: Identification of Bacterial Cell Wall Lyases via Pseudo Amino Acid Composition. Biomed Res Int. 2016; 2016: 1654623. PubMed Abstract | Publisher Full Text | Free Full Text

27. swakkhar: swakkhar/RNAEditing:Pres-a2i (Version v1.1). Zenodo. 2020. http://www.doi.org/10.5281/zenodo.3727221
The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com