Artificial Intelligence In Medicine 113 (2021) 102034

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

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

AIRBP: Accurate identification of RNA-binding proteins using machine learning techniques

Avdesh Mishra [a,1], Reecha Khanal [b,1], Wasi Ul Kabir [b,1], Tamjidul Hoque [b,*]

[a] Department of Electrical Engineering and Computer Science, Texas A&M University-Kingsville, Kingsville, TX, USA
[b] Department of Computer Science, University of New Orleans, New Orleans, LA, USA

ARTICLE INFO

Keywords:
Machine learning
Stacking
RNA-binding proteins
RNA-binding prediction
Protein sequence

ABSTRACT

Identification of RNA-binding proteins (RBPs) that bind to ribonucleic acid molecules is an important problem in Computational Biology and Bioinformatics. It becomes indispensable to identify RBPs as they play crucial roles in post-transcriptional control of RNAs and RNA metabolism as well as have diverse roles in various biological processes such as splicing, mRNA stabilization, mRNA localization, and translation, RNA synthesis, folding-unfolding, modification, processing, and degradation. The existing experimental techniques for identifying RBPs are time-consuming and expensive. Therefore, identifying RBPs directly from the sequence using computational methods can be useful to annotate RBPs and assist the experimental design efficiently. In this work, we present a method called AIRBP, which is designed using an advanced machine learning technique, called stacking, to effectively predict RBPs by utilizing features extracted from evolutionary information, physiochemical properties, and disordered properties. Moreover, our method, AIRBP, use the majority vote from RBPPred, DeepRBPPred, and the stacking model for the prediction for RBPs.

The results show that AIRBP attains Accuracy (ACC), Balanced Accuracy (BACC), F1-score, and Mathews Correlation Coefficient (MCC) of 95.84 %, 94.71 %, 0.928, and 0.899, respectively, based on the training dataset, using 10-fold cross-validation (CV). Further evaluation of AIRBP on independent test set reveals that it achieves ACC, BACC, F1-score, and MCC of 94.36 %, 94.28 %, 0.897, and 0.860, for Human test set; 91.25 %, 93.00 %, 0.896, and 0.835 for S. *cerevisiae* test set; and 90.60 %, 90.41 %, 0.934, and 0.775 for A. *thaliana* test set, respectively. These results indicate that the AIRBP outperforms the existing Deep- and TriPepSVM methods. Therefore, the proposed better-performing AIRBP can be useful for accurate identification and annotation of RBPs directly from the sequence and help gain valuable insight to treat critical diseases.

Availability: Code-data is available here: http://cs.uno.edu/~tamjid/Software/AIRBP/code_data.zip

1. Introduction

RNA Binding Proteins (RBPs) are proteins that bind to ribonucleic acid (RNA) molecules and form dynamic units called ribonucleoprotein (RNP) complexes. These RBPs, along with the RNP complexes, play a crucial role starting from the biogenesis process of RNA to its degradation [1]. Additionally, they contribute to several essential biological functions that include cellular processes (cellular functions, transport, and localization), mRNA stability [2], stress response [3], identifying tumor metastasis signatures [4], tumor differentiation [5], apoptosis, and especially gene regulation at the transcriptional and post-transcriptional levels [6,7]. As an illustration, the newly formed messenger RNA, which carries necessary genetic information from DNA to ribosomes, associates with various RNA binding proteins (RBP) to form messenger ribonucleoprotein (mRNP) complexes [8]. These mRNP complexes govern major elements of the metabolism and functions of mRNA. Similarly, the microRNPs (miRNPs), formed through the association of the RBPs with microRNAs (miRNAs), controls the translation and stability of RNA itself [9]. Identifying RBPs and their mRNA targets are shown useful in cancer therapy [9,10]. Numerous other diseases have been linked to defective RBP expression and functions, such as neuropathies, muscular atrophies, human genetic disorders [11], and metabolic disorders [12]. All this information highlights the urgency of identifying the possible RBPs.

* Corresponding author.
E-mail address: thoque@uno.edu (T. Hoque).
[1] These authors contributed equally to this work as first authors.

https://doi.org/10.1016/j.artmed.2021.102034
Received 11 April 2020; Received in revised form 19 January 2021; Accepted 9 February 2021
Available online 13 February 2021

As of today, numerous studies have been performed, and various experimental and computational methods have been developed to identify and expand our knowledge of RBP. The initial steps towards identification and study of RBPs and RNP complexes date back to almost half a century ago, where experimental methods such as purification of mRNPs from in vitro UV-irradiated polysomal fractions [13], from UV-irradiated intact cells [14], and untreated cells [15] revealed the association of a specific set of proteins with mRNA [8]. Recently, cutting-edge experimental approaches are developed to recognize numerous RBPs, which include the identification of 860 RBPs in human HeLa cells [16] using UV cross-linking methods, 797 RBPs in human embryonic kidney cell line [8] using photoreactive nucleotide-enhanced UV cross-linking and oligo(dT) purification approach, 555 mRNA-binding proteins from mouse embryonic stem cells [17] using UV cross-linking, oligo(dT) and Mass Spectrometry and 120 RBPs from S. *cerevisiae* cells [18] using UV cross-linking and purification methods. Likewise, several RBPs have also been identified from plant cells using the UV cross-linking approach [19–23]. These experiments for identifying and analyzing RBPs, have broadened our understanding of RBPs to a certain extent. Despite the great efforts and achievements, these experiments are expensive, time-consuming, and labor-intensive [24]. Moreover, the tremendous progress in genome sequencing has resulted in an unprecedented amount of genetic information and provided a plethora of protein sequences [25], which outpace the tasks of annotating them and elucidating their functions. Thus, it becomes urgent to have faster and more accurate computational approaches to build an RBP repository and RNA-RBP interaction network maps.

In the recent past, several attempts have been made in identifying RNA-binding proteins, and many effective computational prediction methods have been developed, which can be divided into two broad categories: *i)* templated based; and *ii)* machine learning-based. Template-based methods extract significant structural or sequence similarity between the query and a template known to bind RNA to assess the target sequence's RNA-binding preference [26–28]. Unlike template-based methods, in machine learning methods, the predictive model is created to predict by finding a pattern in the input feature space [29–31]. The machine learning approaches vary in the features employed and the classification algorithm used.

Zhao et al. proposed two template-based approaches for predicting RBPs, of which SPOT-stru [27] is a structure-based approach, and SPOT-seq [26] is a sequence-based approach. In SPOT-stru, the relative structural similarity in the form of Z-score and a statistical energy function DFIRE is used to predict RBPs. The results indicate that SPOT-stru achieved the Mathews Correlation Coefficient (MCC) of 0.57 on the training data of 212 RNA-binding domains and 6761 non-RNA binding domains. On the other hand, in SPOT-seq, the fold recognition between the target sequence and template structures using the defined sequence-structure matching score predicts RBPs. As shown, SPOT-seq achieved the MCC of 0.62 on the training data of 215 RBP chains and 5765 non-binding protein chains.

The machine learning-based approach for predicting RNA-binding proteins involves two crucial steps: *i)* extraction of relevant features and *ii)* selection of an appropriate classification algorithm. Furthermore, depending on the feature extraction mechanism, the existing predictive method can be segmented into two different categories: *i)* extraction of relevant features from the structure of a protein [29,31]; and *ii)* extraction of relevant features from protein sequence [30,32–34]. BindUp [31], available as a web server, is one of the recent structure-based methods that extract electrostatic features and other properties from the protein structure and uses an SVM classifier for RBPs prediction. As reported, BindUp attains sensitivity of 0.71 and specificity of 0.96 on an independent test set of 323 structures of RNA binding proteins and a control set of an equal number extracted from Protein Data Bank (PDB). Towards a sequence-based approach, Ma et al. [32,33] recently proposed two methods, which differ in the features used to train the random forest model for predicting. In [33], the authors incorporated features of evolutionary information combined with physicochemical features (EIPP) and amino acid composition feature to develop the random forest predictor. Besides, in [32], the authors' employed features such as a conjoint triad, binding propensity, non-binding propensity, and EIPP to establish random forest-based predictors with the minimum redundancy maximum relevance (mRMR) method, followed by incremental feature selection (IFS). As reported, their method achieved an accuracy of 0.8662 and MCC of 0.737. Zhang and Liu [34] proposed a new sequence-based approach, namely RBPPred which, integrates the physiochemical properties with the evolutionary information extracted from Position Specific Scoring Matrix (PSSM) profile and utilizes SVM to predict RBPs. As shown, RBPPred correctly predicted 83 % of 2780 RBPs and 96 % of 7093 non-RBPs with MCC of 0.808 using the 10-fold cross-validation (CV) approach. The authors recently proposed an improved deep learning-based method, Deep-RBPPred [35], for predicting RBPs. Deep-RBPPred needs fewer physicochemical properties from the protein sequences and runs much faster compare to RBPPred. Deep-RBPPred is trained on balanced and imbalanced datasets and achieved the MCC of 0.740 and 0.730 on the training data, respectively. Despite significant progress, most of the approaches for RBPs prediction developed in the past are limited in explaining how protein-RNA interactions occur. Thus, it is essential to identify new features, effective encoding technique and advanced machine learning techniques that can help further improve the accuracy of RBPs predictor and ultimately improve our understanding of RNA-protein interactions and their functions.

In this work, we explore different sequence-based features, encoding techniques, and machine learning approaches to further improve RNA-binding proteins' prediction accuracy and our understanding of RNA-protein interaction's binding mechanism. We propose a method, AIRBP, which utilizes features: Evolutionary Information (EI), Physiochemical Properties (PP), and Disordered Properties (DP). It uses four different types of feature encoding technique: Composition, Transition and Distribution (C-T-D) [34], Conjoint Triad (CT) [34,36], PSSM Distance Transformation (PSSM-DT) [37,38] and Residue-wise Contact Energy Matrix Transformation (RCEM-T) [37]. Furthermore, AIRBP utilizes an ensemble machine learning framework, known as stacking [39] and majority voting [40], to predict RBPs from protein sequence only. AIRBP offers a significant improvement in the prediction of RBPs based on the training and independent test datasets when compared to the existing start-of-the-art predictors. Therefore, our predictor can be trusted and used by the research community to guide further the experiments related to RNA-protein interactions and their functions. We believe that the superior performance of AIRBP will motivate the researchers to use it to identify RNA-binding proteins from sequence information. Moreover, the proposed ensemble-based machine learning technique, encoding techniques and features discussed in this work could be applied to tackle other relevant biological problems.

2. Materials and methods

This section describes the approach for training and independent test data preparation, feature extraction and encoding, performance evaluation metrics, and finally, the path we took to establish the ensemble-based machine learning framework for RBPs prediction.

3. Dataset

For this work, we collected the updated version of the training dataset first proposed by [34] from the web link http://rnabinding.com/RBPPred.html. The authors created the updated training dataset [34] from the original training dataset by removing 16 proteins containing RNAs in their crystal structure from the negative set. Therefore, the updated training dataset we collected consists of 7077 non-RBPs (16 proteins removed from the original training dataset, which contained 7093 non-RBPs) and 2780 RBPs (same as the original training dataset).

Next, we found that 13 out of 2780 and 90 out of 7077 protein sequences in RBPs and non-RBPs set, respectively, contained unknown amino acid (X). These sequences containing unknown amino acid (X) were removed from further consideration as the physiochemical properties of an unknown amino acid (X) could not be obtained. After removing the sequences, the training dataset contains 2767 RBPs and 6987 non-RBPs.

Similarly, we also collected the updated version of the test dataset first proposed by [34] from the web link http://rnabinding.com/RBPPred.html. This dataset consists of independent test sets for 3 species, human, S*accharomyces cerevisiae* (S. *cerevisiae* or SC), and A*rabidopsis thaliana* (A. *thaliana* or ATH). The search for the RBP and non-RBPs was made from UniProt and PISCES databases, respectively. Initially, 1551 RBPs and 1350 non-RBPs, 560 RBPs and 395 non-RBPs, and 603 RBPs and 102 non-RBPs were selected for human, S. *cerevisiae,* and A. *thaliana*, respectively. The authors created the test set [34] from the original independent test set by removing 9 proteins from the human set and 7 proteins from S. *cerevisiae* set that had RNAs in their crystal structure from the negative set, respectively. We removed the protein sequences containing unknown amino acid (X) from each of these independent datasets and obtained 967 RBPs and 584 non-RBPs for human, 354 RBPs, and 134 non-RBPs for S. *cerevisiae* and 456 RBPs and 36 non-RBPs for A. *thaliana*.

However, only a few proteins were able to make it into the human, S. *cerevisiae,* and A. *thaliana* category because a significant number of proteins were filtered out either because the relevant features could not be extracted or because the testing and training set both contained identical sequences. Particularly, some proteins were removed as they contain unknown amino acid (X), and no secondary structure or evolutionary information results could be generated for these proteins. Furthermore, additional proteins were removed because these proteins had identical sequences between each of the three testing sets and training set.

We created a new training set and three new independent test sets to evaluate our proposed method from the above train and test sets by removing identical sequences. Identical sequences in train and test sets may lead to bias results. We removed the identical sequence using the CD-HIT tool [41] to ensure that there is no overlap between the training and test set. We combined the training and test set and ran the CD-HIT tool with the sequence identity cutoff of 25 % and collected a training set that includes 2642 RBPs and 6884 non-RBPs and three independent test sets that contain a total of 51 RBPs and 144 non-RBPs for human, 30 RBPs and 50 non-RBPs for S. c*erevisiae,* and 48 RBPs and 14 non-RBPs for A. *thaliana*.

Moreover, to increase the number of proteins in A. *thaliana* test set, we further downloaded 836 A. *thaliana* RBPs recently published by Marondedze [42]. To balance the new data's positive and negative samples, we downloaded an additional 1368 non-RBPs from Protein Data Bank (PDB). Further, we utilized the same techniques used to create the negative samples (non-RBPs) in the RBPpred [34] method, to obtain a non-redundant non-RBPs set. Subsequently, we combined the new data with the training and test set mentioned above and reran the CD-HIT tool with the sequence identity cutoff of 25 % to obtain 301 RBPs and 128 non-RBPs for the new A. *thaliana* dataset. Then, we added 80 % of the new A. *thaliana* data to the training set and the remaining 20 % data to the test set. Finally, we obtained a training set that includes 2882 RBPs and 6986 non-RBPs and A. *thaliana* test set that contains 109 RBPs and 40 non-RBPs. The human and S. *cerevisiae* test dataset remained unchanged.

3.1. Balanced training dataset

The training dataset contains 2642 RBPs and 6884 non-RBPs, which is highly imbalanced. The imbalanced problem can be mitigated by undersampling or oversampling the dataset. Oversampling methods create new synthetic examples in the minority class, whereas undersampling methods delete or merge examples in the majority class. Undersampling can remove important data points. Therefore, we choose an oversampling method, Synthetic Minority Oversampling Technique (SMOTE), to make the dataset balanced. SMOTE is a widely used oversampling method [43]. Smote creates synthetic minority class samples by generating new samples on the lines connecting a point (sample) and one of its K-nearest neighbors. We extracted the probabilities from the base classifiers with the balanced dataset using SMOTE. The meta classifier is trained with the imbalanced dataset.

3.2. Feature extraction

To create an effective RBPs predictor from sequence alone, the feature vector for each protein sequence was derived from the PSSM profile, Physiochemical Properties (PP), Residue-wise Contact Energy Matrix (RCEM), and Molecular Recognition Features (MoRFs). A total of 10 different properties was encoded with a vector of 2603 dimensions to represent a protein sequence, as shown in Fig. 1. Out of 10, five distinct properties: hydrophobicity, polarity, normalized van der Waals volume, polarizability, and predicted secondary structure that belongs to the PP group were each encoded via 21 dimension vector utilizing the C—T—D encoding technique [44,45]. Moreover, the remaining five properties, solvent accessibility, charge, and polarity of the side chain, MoRFs, RCEM, and PSSM profile, were encoded via 13, 64, 1, 20, and 2400 dimensional vectors, respectively. Here, PSSM belongs to the EI group, and MoRFs and RCEM belong to the DP group. The properties, solvent accessibility, charge, and polarity of the side chain, RCEM, and PSSM profile were encoded utilizing C—T—D, CT [34,36], RCEM transformation [37], and PSSM-DT transformation techniques [37,38], respectively. Each of the 10 properties, along with their encoding mechanism, is described next in detail.

3.3. Features extracted from physicochemical properties

In this section, we describe various feature extraction techniques we utilized to obtain a fixed dimensional feature vector from the physicochemical properties, which include hydrophobicity, polarity, normalized van der Waals volume, polarizability, predicted secondary structure, solvent accessibility, and charge and polarity of the side chain to encode protein sequence.

3.4. Composition, transition and distribution (C—T—D) transformation features

In this section, the C—T—D transformation method aims to describe the distribution patterns of amino acid properties. This method to compute distribution patterns of amino acid properties was first suggested by [49] for protein fold class prediction. In our implementation, we used C—T—D transformation to encode the properties, including hydrophobicity, polarity, normalized van der Waals volume, polarizability, predicted the secondary structure, and solvent accessibility. As the name suggests, this transformation technique focuses on three different components: composition of a particular amino acid in the sequence, a transition of one amino acid to another as we go linearly through the sequence, and distribution referring to how one amino acid group is distributed throughout the protein sequence [50,51]. To create a consistent number of features for proteins with different sequence lengths, 20 standard amino acids are divided into 3 groups [52] based on their hydrophobicity, normalized van der Waals volume, polarity, and polarizability. Fig. 2 illustrates the C—T—D transformation technique while the 20 standard amino acids are divided into 2 groups which, generates a feature vector of 13 dimensions. Following the transformation similar to Fig. 2 but with amino acids classified into 3 groups rather than 2, we obtain a feature vector of 21 dimensions for the physiochemical properties such as hydrophobicity, normalized van der Waals volume, polarity, and polarizability.

Furthermore, to encode the predicted secondary structure and

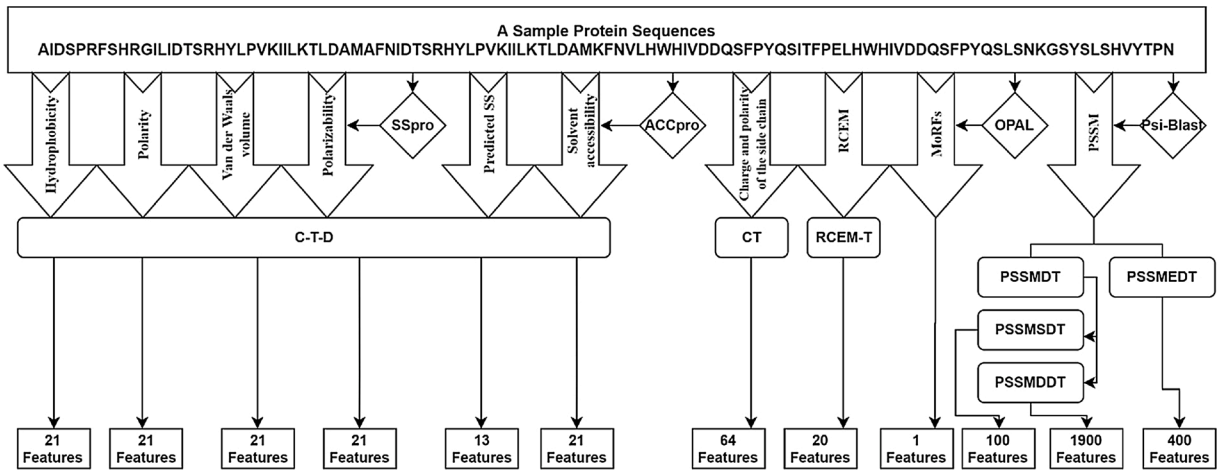

Fig. 1. Illustration of encoding the protein sequence into a feature vector of 2603 features utilizing various feature encoding technique. Here, the predicted SS and surface accessibility were obtained from SSpro and ACCpro program [46]. Likewise, the MoRFs scores were predicted using the OPAL program [47], and the PSSM scores were obtained using the PSI-BLAST program [48].

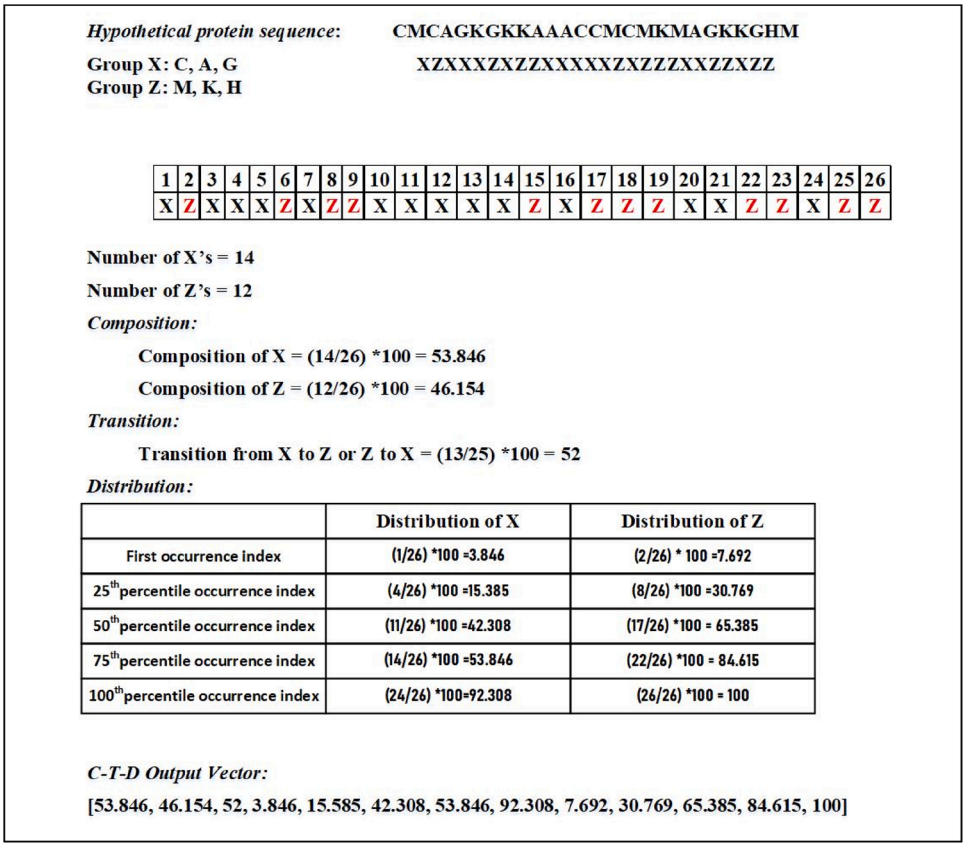

Fig. 2. Illustration of the C—T—D transformation technique. The 20 standard amino acids are divided into 2 groups (e.g., X and Z). First, the group index (X or Z) of every amino acid in the protein sequence is extracted, and consequently, a vector of 13 dimensions is obtained through composition, transition, and distribution.

solvent accessibility as features, we first used the SSpro and ACCpro program [46] to predict secondary structure in the form of 'H' (helix), 'E' (strand), and 'C' (other than helix and strand) and solvent accessibility in the form of 'e' (exposed residues) and '-' (buried residues), respectively. The choice of SSpro and ACCpro was made to extract predicted secondary structure and solvent accessibility because of its superior performance and remarkable speed. As reported, SSpro and ACCpro [46] achieved an accuracy of 92.9 % and 90 % for secondary structure prediction and relative solvent accessibility prediction, respectively. Using the transformation technique described above, we obtained a feature vector of 21 dimensional and 13 dimensions for predicted secondary structure and solvent accessibility, respectively.

3.5. Conjoint triad (CT) transformation features

While the 20 standard amino acids are divided into 4 groups (Group A, B, C, and D representing acidic, basic, polar, and non-polar, respectively), Shen et al. first proposed the CT transformation technique for

protein-protein interaction prediction [53], which was successfully applied for protein-RNA interaction prediction in the past [34,36]. In our implementation, we adopted the CT transformation technique to encode the protein sequence based on the side chain's charge and polarity of the amino acids in a protein. First, the 20 standard amino acids are divided into 4 groups: *i)* acidic (contain residues D and E); *ii)* basic (contain residues H, R and K); *iii)* polar (contain residues C, G, N, Q, S, T, and Y); and *iv)* non-polar (contain residues A, F, I, L, M, P, V, and W) according to their charge and polarity of the side chain. Then, the protein sequence is converted into a sequence of group types where each element in the sequence represents a group type of the corresponding amino acid in the protein sequence. Next, a triad of three contiguous amino acids is considered as a single unit. Accordingly, all the triads can be classified into $4 \times 4 \times 4 = 64$ classes. Finally, a sliding window of a triad is passed through a sequence of group types, and the frequency of occurrences of each type of triad is counted. We obtain a feature vector of 64 dimensions for charge and polarity of side chains of amino acids in a protein through this process. Fig. 3 illustrates the CT transformation technique we used to extract features from protein sequences based on side chains' charge and polarity.

3.6. Features extracted from evolutionary information

This section describes various feature extraction techniques utilized to obtain a fixed dimensional feature vector from the evolutionary information, called PSSM profile to encode protein sequence.

Evolutionary information is one of the most critical information useful for solving various biological problems and has been widely used in many research work [34,54–58]. In this work, the evolutionary information in the form of the PSSM profile is directly obtained from the protein sequence and later transformed into a fixed dimensional vector. PSSM captures the conservation pattern in multiple alignments and preserves it as a matrix for each position in the alignment. The high score in the PSSM matrix indicates more conserved positions, and the lower score indicates less conserved positions [57]. For this study, we generated the PSSM profile for every protein sequence by executing three iterations of PSI-BLAST against NCBI's non-redundant database [48]. The evolutionary information in the PSSM profile is represented as a matrix of L $\times$ 20 dimensions, where L is the length of the protein sequence. A particular element $M_{i,j}$ of the PSSM matrix, represents the occurrence probability of the amino acid i at position j of a protein sequence.

Particularly, we apply Position Specific Scoring Matrix Distance Transform (PSSM-DT) that includes PSSM-Similar Distance Transform (PSSM-SDT) and PSSM-Different Distance Transform (PSSM-DDT) to extract occurrence probabilities for the pairs of same amino acids and different amino acids, respectively as features. Moreover, we apply Evolutionary Distance Transform (EDT) to extract the non-co-occurrence probability of two amino acids as features. Here, the evolutionary information is inherently captured in the form of a PSSM profile as PSI-BLAST generates PSSM profile for target protein sequence by performing multi-sequence alignment with the NCBI's non-redundant database. In general, multi-sequence alignment provides information about how close or distinct the target protein is with the proteins in the large NCBI database, which contains protein sequences of species that have evolved with time. Moreover, it provides information regarding the conserved region of a target protein sequence.

3.7. PSSM-Distance transformation (PSSM-DT) features

We use two types of distance transformation techniques [37,38]: *i)* the PSSM distance transformation for the same pairs of amino acids

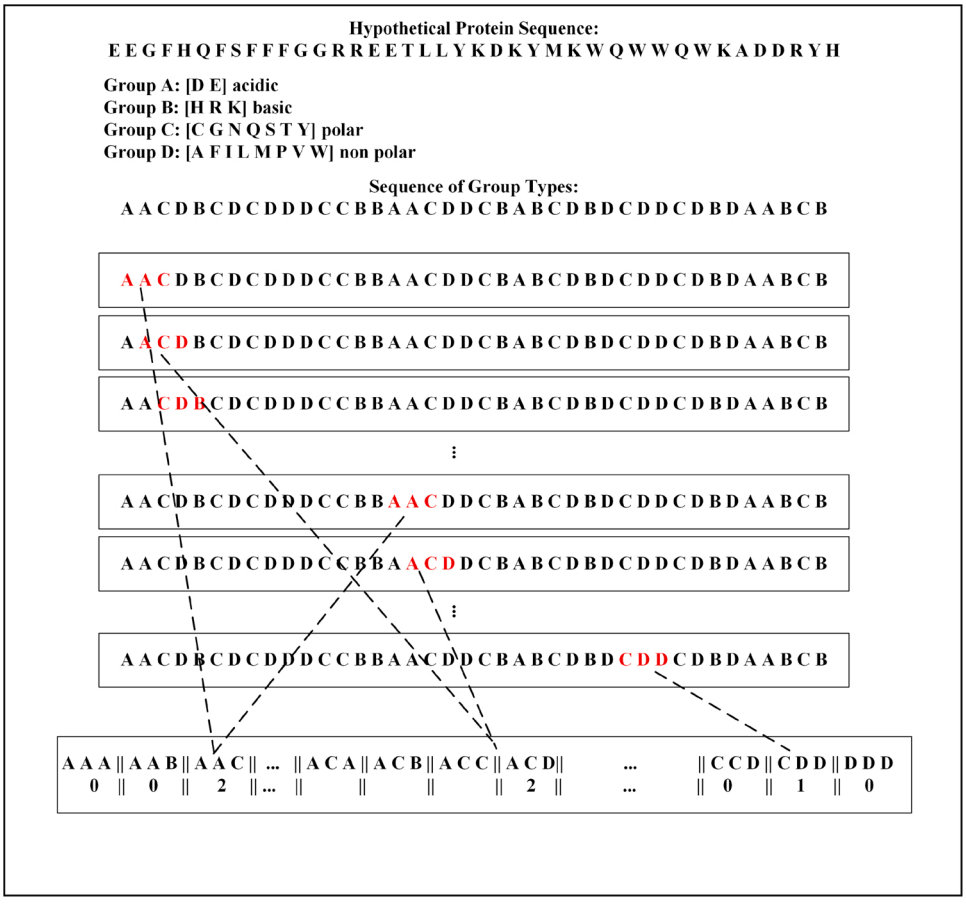

Fig. 3. Illustration of Conjoint Triad transformation technique.

(PSSM-SDT); and *ii)* the PSSM distance transformation for different pairs of amino acids (PSSM-DDT), together known as PSSM-DT to extract fixed dimensional feature vectors of size 100 and 1900, respectively.

Utilizing PSSM-SDT, we compute the occurrence probabilities for the pairs of the same amino acids separated by a distance D along the sequence, which can be represented as:

$$PSSM - SDT(j,\ D) = \sum_{i=1}^{L-D} M_{i,j} * M_{i+D,\,j} \Big/ (L - D) \tag{1}$$

where j represents one type of the amino acid, L represents the length of the sequence, $M_{i,j}$ represents the PSSM score of amino acid j at position i, and $M_{i+D,j}$ represents the PSSM score of amino acid j at position $i + D$. Through this approach, $20 \times K$ number of features were generated where K is the maximum range of D ($D = 1,2, \ldots, K$).

Likewise, utilizing PSSM-DDT, we compute the occurrence probabilities for pairs of different amino acids separated by a distance D along the sequence, which can be represented as:

$$PSSM - DDT(i_1, i_2,\ D) = \sum_{j=1}^{L-D} M_{j,i_1} * M_{j+D,i_2} \Big/ (L - D) \tag{2}$$

where, i_1 and i_2 represent two different types of amino acids. The total number of features obtained by PSSM-DDT is $380 \times K$. Here, we consider $K = 5$. Therefore, 100 features were obtained by PSSM-SDT, and PSSM-DDT transformation techniques obtained a total of 1900 features.

3.8. Evolutionary distance transformation (EDT) features

Unlike PSSM-DT, the EDT approximately measures the non-co-occurrence probability of two amino acids separated by a specific distance d in a protein sequence from the PSSM profile [57,59]. The EDT is calculated from the PSSM profile as:

$$f(R_x, R_y) = \sum_{d=1}^{D} \frac{1}{L-d} \sum_{i=1}^{L-d} \left(M_{i,x} - M_{i+d,y}\right)^2 \tag{3}$$

where d is the distance separating two amino acids, D is the maximum value of d, $M_{i,x}$ and $M_{i+d,y}$ are the elements in the PSSM profile, and R_x and R_y represent any of the 20 standard amino acids in the protein sequence. Here, the value of $D = L_{min}\text{-}1$ where L_{min} is the length of the shortest protein sequence in the training dataset. Using EDT, we obtain a feature vector of dimension 400.

3.9. Features extracted from disordered properties

This section describes a feature extraction technique utilized to obtain a fixed dimensional feature vector from the residue-wise contact energy matrix to encode protein sequence.

RBPs are found to bind with RNA through classically structured RNA-binding domains and intrinsically disordered regions (IDRs) [60]. For example, approximately 20 % of the identified mammalian RBPs (~170 proteins) were found to be disordered by over 80 % [61]. The energy contribution of a large number of inter and intra-residual interactions in intrinsically disordered proteins (IDPs) cannot be approximated by the energy functions extracted from known structures [58,62–65] as IDPs lack a defined and ordered 3D structure [66]. Therefore, to inherently incorporate important information regarding the IDRs and amino acid interactions, we employed the predicted residue-wise contact energies [67] and molecular recognition features (MoRFs) [47] to encode the protein sequence.

3.10. Residue-wise contact energy matrix transformation (RCEM-T) features

We adopted the predicted residue-wise contact energy matrix (RCEM) derived in [67], by the least square fitting of 674 proteins primary sequence with the contact energies derived from the tertiary structure of 785 proteins. As shown in Table 1, the RCEM is a 20×20 dimensional matrix that contains residue-wise contact energy for 20 standard amino acids. For a protein sequence of length L, an $L \times 20$ dimensional matrix M is obtained, which holds a 20-dimensional vector for each amino acid in a protein sequence. The resulting matrix M is then encoded into a feature vector of 20 dimensional by computing the column-wise sum as:

$$f(A_j) = \sum_{i=1}^{L} m_{i,j}\ (j = 1, 2, \cdots,\ 20) \tag{4}$$

where $m_{i,j}$ is the element of matrix M, i is the amino acid index in a sequence, and j represents 20 standard amino acid types. The final feature vector, $RCEM - T = [v_1,\ v_2,\ \cdots,\ v_{20}]$ is obtained by dividing each element in *RCEM-T* by the sum of all the elements in the same vector. Considering V_s as the sum of all the elements in the RCEM-T vector, each component of the final *RCEM-T* vector can be represented as:

$$RCEMT(v_i) = \frac{v_i}{V_s} \tag{5}$$

3.11. Molecular recognition features (MoRFs)

MoRFs, also known as molecular recognition elements (MoREs), are disordered regions in a protein that exhibit various molecular recognition and binding functions [68]. Post-translational modifications (PTMs) can induce disorder to order transitions of IDPs upon binding with their binding partners, which could be either RNA, DNA, proteins, lipids, carbohydrates, or other small molecules [69,70]. MoRFs play a vital role in IDPs' various biological functions located within long disordered protein sequences [47,71–73]. Additionally, Mohan et al. suggest that functionally significant residual structures exist in MoRF regions before the actual binding [74]. These residual structures could, therefore, be useful in the prediction of binding between proteins and RNA. Here, to capture the functional properties of IDRs that may bind to RNAs, we employ a single predicted MoRFs score as a feature. To obtain a single predicted MoRFs score, first, the residue-wise predicted MoRFs scores are obtained from the OPAL program [47]. A single predicted MoRFs score is computed by taking a ratio of the sum of the residue-wise MoRFs score and the length of the protein sequence.

3.12. Performance evaluation

To evaluate the performance of AIRBP, we adopted a widely used 10-fold CV and the independent testing approach. In the process of 10-fold CV, the dataset is segmented into 10 parts, which are each of about the same size. When one fold is kept aside for testing, the remaining 9 folds are used to train the classifier. This training and test process is repeated until each fold has been kept aside once for testing, and consequently, the test accuracy of each fold is combined to compute the average [75]. Unlike a 10-fold CV, in independent testing, the classifier is trained with the training dataset and consequently tested using the independent test dataset. Independent testing ensures that none of the samples in the independent test set are present in the training dataset. We used several performance evaluation metrics listed in Table 2 and ROC and AUC to test the performance of the proposed method and compare it with the existing approaches. AUC is the area under the receiver operating characteristics (ROC) curve, which is used to evaluate how well a predictor separates two classes of information (RNA-binding and non-binding protein).

Table 1
RCEM table is used to obtain RCEM-T features.

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	−1.65	−2.83	1.16	1.8	−3.73	−0.41	1.9	−3.69	0.49	−3.01	−2.08	0.66	1.54	1.2	0.98	−0.08	0.46	−2.31	0.32	−4.62
C	−2.83	−39.58	−0.82	−0.53	−3.07	−2.96	−4.98	0.34	−1.38	−2.15	1.43	−4.18	−2.13	−2.91	−0.41	−2.33	−1.84	−0.16	4.26	−4 .46
D	1.16	−0.82	0.84	1.97	−0.92	0.88	−1.07	0.68	−1.93	0.23	0.61	0.32	3.31	2.67	−2.02	0.91	−0.65	0.94	−0.71	0. 90
E	1.8	−0.53	1.97	1.45	0.94	1.31	0.61	1.3	−2.51	1.14	2.53	0.2	1.44	0.1	−3.13	0.81	1.54	0.12	−1.07	1. 29
F	−3.73	−3.07	−0.92	0.94	−11.25	0.35	−3.57	−5.88	−0.82	−8.59	−5.34	0.73	0.32	0.77	−0.4	−2.22	0.11	−7.05	−7.09	−8 .80
G	−0.41	−2.96	0.88	1.31	0.35	−0.2	1.09	−0.65	−0.16	−0.55	−0.52	−0.32	2.25	1.11	0.84	0.71	0.59	−0.38	1.69	−1 .90
H	1.9	−4.98	−1.07	0.61	−3.57	1.09	1.97	−0.71	2.89	−0.86	−0.75	1.84	0.35	2.64	2.05	0.82	−0.01	0.27	−7.58	−3 .20
I	−3.69	0.34	0.68	1.3	−5.88	−0.65	−0.71	−6.74	−0.01	−9.01	−3.62	−0.07	0.12	−0.18	0.19	−0.15	0.63	−6.54	−3.78	−5 .26
K	0.49	−1.38	−1.93	−2.51	−0.82	−0.16	2.89	−0.01	1.24	0.49	1.61	1.12	0.51	0.43	2.34	0.19	−1.11	0.19	0.02	−1 .19
L	−3.01	−2.15	0.23	1.14	−8.59	−0.55	−0.86	−9.01	0.49	−6.37	−2.88	0.97	1.81	−0.58	−0.6	−0.41	0.72	−5.43	−8.31	−4 .90
M	−2.08	1.43	0.61	2.53	−5.34	−0.52	−0.75	−3.62	1.61	−2.88	−6.49	0.21	0.75	1.9	2.09	1.39	0.63	−2.59	−6.88	−9 .73
N	0.66	−4.18	0.32	2.2	0.73	−0.32	1.84	−0.07	1.12	0.97	0.21	0.61	1.15	1.28	1.08	0.29	0.46	0.93	−0.74	0. 93
P	1.54	−2.13	3.31	1.44	0.32	2.25	0.35	0.12	0.51	1.81	0.75	1.15	−0.42	2.97	1.06	1.12	1.65	0.38	−2.06	−2 .09
Q	1.2	−2.91	2.67	0.1	0.77	1.11	2.64	−0.18	0.43	−0.58	1.9	1.28	2.97	−1.54	0.91	0.85	−0.07	−1.91	−0.76	0. 01
R	0.98	−0.41	−2.02	−3.13	−0.4	0.84	2.05	0.19	2.34	−0.6	2.09	1.08	1.06	0.91	0.21	0.95	0.98	0.08	−5.89	0. 36
S	−0.08	−2.33	0.91	0.81	−2.22	0.71	0.82	−0.15	0.19	−0.41	1.39	0.29	1.12	0.85	0.95	−0.48	−0.06	0.13	−3.03	−0 .82
T	0.46	−1.84	−0.65	1.54	0.11	0.59	−0.01	0.63	−1.11	0.72	0.63	0.46	1.65	−0.07	0.98	−0.06	−0.96	1.14	−0.65	−0 .37
V	−2.31	−0.16	0.94	0.12	−7.05	−0.38	0.27	−6.54	0.19	−5.43	−2.59	0.93	0.38	−1.91	0.08	0.13	1.14	−4.82	−2.13	−3 .59
W	0.32	4.26	−0.71	−1.07	−7.09	1.69	−7.58	−3.78	0.02	−8.31	−6.88	−0.74	−2.06	−0.76	−5.89	−3.03	−0.65	−2.13	−1.73	−1 2.39
Y	−4.62	−4.46	0.9	1.29	−8.8	−1.9	−3.2	−5.26	−1.19	−4.9	−9.73	0.93	−2.09	0.01	0.36	−0.82	−0.37	−3.59	−12.39	−2 .68

Table 2
Name and definition of the evaluation metric.

Name of Metric	Definition
True Positive (TP)	Correctly predicted RNA-binding proteins
True Negative (TN)	Correctly predicted non-RNA-binding proteins
False Positive (FP)	Incorrectly predicted RNA-binding proteins
False Negative (FN)	Incorrectly predicted non-RNA-binding proteins
Recall/Sensitivity/True Positive Rate (SN)	$\frac{TP}{TP+FN}$
Specificity/True Negative Rate (SP)	$\frac{TN}{TN+FP}$
Fall Out Rate /False Positive Rate (FPR)	$\frac{FP}{FP+TN}$
Miss Rate/False Negative Rate (FNR)	$\frac{FN}{FN+TP}$
Accuracy (ACC)	$\frac{TP+TN}{FP+TP+TN+FN}$
Balanced Accuracy (BACC)	$\frac{1}{2}\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right)$
Precision (PR)	$\frac{TP}{TP+FP}$
F1-score (Harmonic mean of precision and recall)	$\frac{2TP}{2TP+FP+FN}$
Mathews Correlation Coefficient (MCC)	$\frac{(TPTN)-(FPFN)}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}}$

3.13. Framework of AIRBP

To develop the AIRBP predictor for RBPs prediction, we adopted an idea of a stacking [39] and majority voting [40] based machine learning approach, which has recently been successfully applied to solve various bioinformatics problems [37,76–78]. Stacking is an ensemble-based machine learning approach, which collects information from multiple models in different phases and combines them to form a new model. Stacking is considered to yield more accurate results than the individual machine learning methods as the information gained from more than one predictive model minimizes the generalization error. The stacking method includes two-levels of classifiers, where the classifiers of the first-level are called base-classifiers, and the classifiers of the second-level are called meta-classifiers. In the first level, a set of base-classifiers C_1, C_2, …, C_N is employed [79]. The prediction probabilities from the base-classifiers are combined using a meta-classifier to reduce the generalization error and improve the accuracy of the predictor. To enrich the meta-classifier, with necessary information on the problem space, the classifiers at the base-level are selected, such that their underlying operating principle is different from one another [37, 78]. Majority Voting is also an ensemble machine learning algorithm [40] that involves summing the class labels' votes from different models and predicting the class based on the majority votes.

To select the classifiers to use in the first and second level of the AIRBP stacking method, we analyzed the performance of seven individual classification methods: *i)* Random Decision Forest (RDF) [80]; *ii)* Bagging (Bag) [81]; *iii)* Extra Tree (ET) [82]; iv) Extreme Gradient Boosting (XGBoost or XGB) [83]; v) Logistic Regression (LogReg) [75, 84]; vi) K-Nearest Neighbor (KNN) [85]; and LightGBM [86]. The algorithms and their configuration details are briefly discussed below.

i) RDF: RDF [80] constructs many decision trees, each of which is trained on a random subset of the training data. The sub-set used to create a decision tree is constructed from a given set of observations of training data by taking 'm' observations at random and with replacement (a.k.a. Bootstrap Sampling). Next, the final predictions are achieved by aggregating the prediction from the individual decision trees. For classification, the final prediction is made by computing the mode (the value that appears most often) of the classes (in our case: whether a protein is RNA-binding or non-binding). In our implementation of the RDF, we used bootstrap samples to construct 1000 trees (n_estimators = 1000) in the forest, and the rest of the parameters were set to their default value.

ii) Bag: Bag [81] machine learning algorithm operates by forming a class of algorithms that creates several instances of a base classifier/-estimator on random subsets of the training samples and consequently combines individual predictions to yield a final prediction. It reduces the variance in the prediction. In our study, the BAG classifier was fit on multiple subsets of data using Bootstrap Sampling using 1000 decision trees (n_estimators = 1000), and the rest of the parameters were set to their default value.

iii) ET: Extremely randomized tree (ET) classifier [82] operates by fitting several randomized decision trees (a.k.a. extra-trees) on various sub-sets and uses averaging to improve the prediction accuracy and control over-fitting. In our implementation, the ETC model was constructed using 1000 trees (n_estimators = 1000), and the Gini impurity index assessed the quality of a split. The rest of the parameters were set to their default value.

iv) XGBoost: XGBoost [83] follows the same principle of gradient boosting as the Gradient Boosting Classifier (GBC). GBC (Friedman, 2001) involves three elements: (a) a loss function to be optimized, (b) a weak learner to make predictions, and (c) an additive model to add weak learners to minimize the loss function. GBC's objective is to minimize the loss of the model by adding weak learners in a stage-wise fashion using a procedure similar to gradient descent. The existing weak learners in the model remain unchanged while adding new weak learners. The new learner's output is added to the output of the existing sequence of learners to correct or improve the model's final output. Unlike GBC, XGBoost performs more regularized model formalization to control over-fitting, which results in better performance. In addition to increased performance, XGBoost provides higher computational speed. In our configuration of XGBoost, the values of parameters: colsample_-bytree, gamma, min_child_weight, learning_rate, max_depth, n_estimators, and subsample ratio were optimized to achieve the best 10-fold cross-validation accuracy using a grid search [87] technique. The best values of the parameters: colsample_bytree, gamma, min_child_weight, learning_rate, max_depth, n_estimators, and subsample ratio were found to be 0.6, 0.3, 1.5, 0.07, 5, 10,000 and 0.95, respectively. And the rest of the parameters were set to their default value.

v) LogReg: LogReg (a.k.a. logit or MaxEnt) [75,84] is a machine learning classifier that measures the relationship between the categorical dependent variable (in our case: an RNA-binding or non-binding proteins) and one or more independent variables by generating an estimation probability using logistic regression. In our implementation, we set all the parameters of LogReg to their default values.

vi) KNN: KNN [85] is a non-parametric and lazy learning algorithm. Non-parametric means it does not make any assumption for underlying data distribution, instead, it creates models directly from the dataset. Furthermore, lazy learning means it does not need any training data points for a model generation rather uses the training data while testing. It works by learning from the K number of training samples closest in the distance to the target point in the feature space. The classification decision is made based on the majority-votes obtained from the K nearest neighbors. Here, we set the value of K to 9 and the rest of the parameters to their default value.

vi) LightGBM: LightGBM [86] also follows the gradient boosting framework that uses tree-based learning algorithms. The algorithm has a faster training speed, higher efficiency, and lower memory usage compared to XGBoost. It also supports parallel and GPU learning and capable of handling large-scale data. In our implementation, the LightGBM model was constructed using 1000 trees (n_estimators = 1000), and the rest of the parameters were set to their default value.

All the classification methods mentioned above are built and optimized using python's Scikit-learn library [88]. To design a stacking method for AIRBP, we evaluated the different combinations of base-classifiers and finally selected the one that provided the highest performance.

The set of stacking method tested are:

i) SF1: RDF, LightGBM, LogReg, KNN in base-level and XGBoost in meta-level,
ii) SF2: ETC, LightGBM, LogReg, KNN in base-level, and XGBoost in meta-level.
iii) SF3: Bag, LightGBM, LogReg, KNN in base-level, and XGBoost in meta-level.

Here, the choice of base-level classifiers is made such that the underlying principle of learning of each of the classifiers is different from each other [37]. For example, in SF1, SF2, and SF3, the tree-based classifiers RDF, Bag, and ET are individually combined with the other two methods LogReg and KNN, to learn different information from the problem-space. Additionally, for each of the combinations SF1, SF2, and SF3, the XGBoost classifier is used in the meta-level, and LightGBM is used in the base-level because they performed well among all the other individual methods applied in this work. Moreover, the use of the combination of XGBoost and LightGBM provides us more information about the problem-space. It also improves time efficiency because LightGBM is faster than XGBoost. While examining the 10-fold CVs performance of the above three combinations, we found that the second stacking method, SF2 attains the highest performance. Therefore, we employ four classifiers ETC, LightGBM, LogReg, and KNN, as the base classifiers and XGBoost as the AIRBP stacking method's meta-classifier. In AIRBP, the probabilities of both the classes (RBP and non-RBP) generated by the four base-classifiers are combined with the features and provided as input features to the meta-classifier. Moreover, to achieve better performance, the probabilities are extracted from the base classifiers with a balanced dataset, which is created using SMOTE, and the meta classifier is trained with the imbalanced dataset. Finally, we take the majority vote from RBPPred [34], the imbalance Deep-RBPPred model [35], and the stacking model for the prediction for RBPs. We chose the imbalance Deep-RBPPred model in majority voting as our final model is trained with the imbalance dataset. Fig. 4 shows the prediction framework of the AIRBP.

4. Results

In this section, we first demonstrate the performance comparison of potential base-classifiers and the stacking method. Finally, we report the performance of AIRBP on the training dataset and three independent test datasets and, consequently, compare it with the existing methods.

4.1. Selection of classifiers for stacking

To select the methods to use as the base and the meta-classifiers, we analyzed the performance of seven different machine learning algorithms: LightGBM, RDF, Bag, ET, XGBoost, LogReg, and KNN on the training dataset through a 10-fold CV approach. The performance comparison of the individual classifiers on the training dataset is shown in Table 3.

Table 3 further shows that the XGBoost and LightGBM are the best performing classifier among seven different classifiers implemented in our study in terms of balanced accuracy, accuracy, F1-score, and MCC. Moreover, the LightGBM attains balanced accuracy, accuracy, F1-score, and MCC of 93.65 %, 95.01 %, 0.914, and 0.879, respectively, whereas XGBoost attains balanced accuracy, accuracy, F1-score, and MCC of 93.21 %, 94.61 %, 0.907, and 0.869, respectively. Furthermore, it is evident from Table 3 that the balanced accuracy, accuracy, F1-score, and MCC of the XGBoost and LightGBM are higher than ET, RDF, LogReg, KNN, and Bag, respectively. The XGBoost and LightGBM algorithms' greater performance motivated us to use them both in base and meta-classier in the AIRBP prediction framework.

To further select the classifiers to be used at the base-level, we adopted base-classifier selection guidelines based on different underlying principles. Therefore, we used KNN and LogReg as two additional classifiers at the base-level. Then, we added a single tree-based ensemble

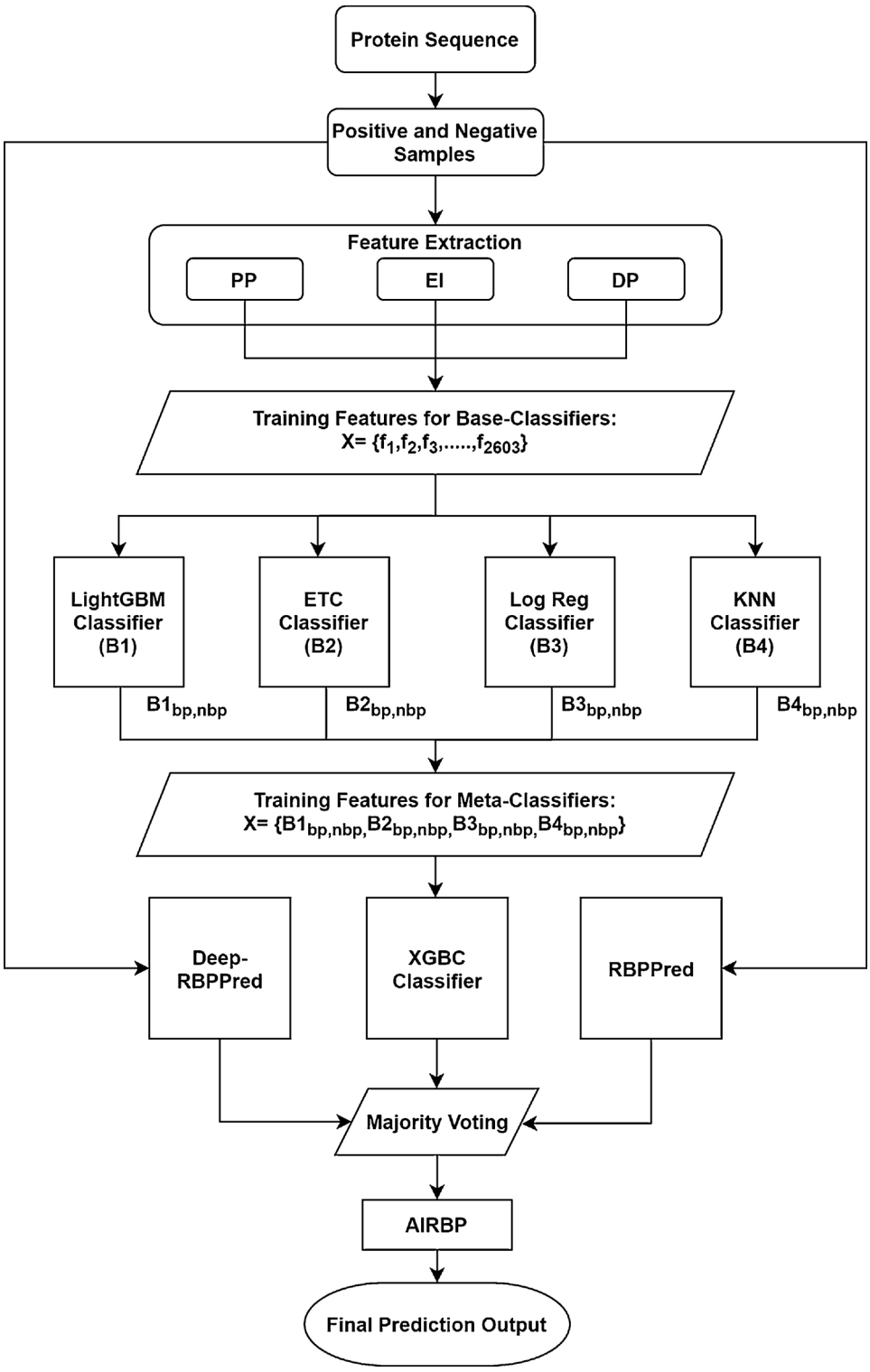

Fig. 4. Prediction framework of the AIRBP.

method out of three methods, RDF, Bag, and ET, at a time as the fourth base-classifier and designed three different combinations of stacking method, namely SF1, SF2, and SF3. We added LightGBM as a base classifier and XGBoost as the meta classifier as they are the better performing among the other classifiers. The performance comparison of SF1, SF2, and SF3 stacking method on the training dataset using 10-fold CV are presented in Table 4. Table 4 demonstrates that SF2 outperforms both SF1 and SF3. Table 4 shows that SF2 has higher balanced accuracy, accuracy, F1-score, and MCC compared to SF1 and SF3. Hence, we select SF2, including ETC, LightGBM, LogReg, and KNN as base-classifiers and another XGBoost as a meta-classifier, as our final predictor.

4.2. Performance comparison with existing approaches on the training dataset

Here, we compare the performance of AIRBP with Deep-RBPPred [35] on the training dataset using the 10-fold CV approach. Deep-RBPPred and TriPepSVM [89] were recently proposed to predict

Table 3
Comparison of various machine learning algorithms on the training dataset through a 10-fold CV.

Metric/Methods	LightGBM	KNN	LogReg	Bag	RDF	XGB	ETC
SN (%)	90.35	**92.26**	91.85	88.45	81.37	89.83	76.79
SP (%)	96.94	69.78	93.34	95.32	96.99	96.58	**97.42**
BACC (%)	**93.65**	81.02	92.59	91.88	89.18	93.21	87.11
ACC (%)	**95.01**	76.35	92.91	93.31	92.43	94.61	91.40
FPR	0.031	0.302	0.067	0.047	0.030	0.034	**0.026**
FNR	0.096	**0.077**	0.082	0.116	0.186	0.102	0.232
PR (%)	92.41	55.74	85.06	88.63	91.78	91.55	**92.48**
F1-score	**0.914**	0.695	0.883	0.885	0.863	0.907	0.839
MCC	**0.879**	0.565	0.834	0.838	0.813	0.869	0.787

The best score values are **bold**faced.

Table 4
Comparison of different stacking method with a different set of base-classifiers on the training dataset through a 10-fold CV.

Metrics / Stacking Model	SF1	SF2	SF3
SN (%)	91.57	**92.02**	91.53
SP (%)	97.19	**97.41**	97.18
BACC (%)	94.38	**94.71**	94.36
ACC (%)	95.55	**95.84**	95.53
FPR	0.028	**0.026**	0.028
FNR	0.084	**0.080**	0.085
PR (%)	93.09	**93.61**	93.05
F1-score	0.923	**0.928**	0.923
MCC	0.892	**0.899**	0.891

The best score values are **bold**faced.

RBPs directly from the sequence and have been shown to yield better predictions than other existing approaches. We did not compare AIRBP with the TriPepSVM method as TriPepSVM is not a generic method that can be applied to any species. Furthermore, it is to be noted that AIRBP uses the subset of the training dataset used in Deep-RBPPred as we have removed the identical sequences from the training and test dataset given by Zhang et al. [34]. For the comparison, the quantities for all the evaluation metrics for Deep-RBPPred are obtained from Zheng et al. [35]. The prediction results of AIRBP and Deep-RBPPred on the training dataset computed using 10-fold CV are listed in Table 5.

From Table 5, we observed that AIRBP outperforms the Deep-RBPPred imbalance and balance model based on the MCC metric. We report MCC's value only for the Deep-RBPPred predictor as Deep-RBPPred did not report the other metric values for the training dataset. AIRBP provides a 23.15 % and 21.49 % improvement over the imbalance Deep-RBPPred and balanced Deep-RBPPred model, respectively, based on the MCC metric.

MCC considers true and false positives and negatives and is generally considered a balanced measure that can be used even though the classes are of very different sizes. From Table 5, it is clear that based on MCC, AIRBP outperforms both balance and imbalance Deep-RBPPred methods.

4.3. Performance comparison with existing approaches on the independent test set

In this section, we compare the performance of AIRBP with two recent predictors, Deep-RBPPred [35] and TriPepSVM [89], on three different independent test sets, Human, S. *cerevisiae,* and A. *thaliana*. To compare, we ran both Deep-RBPPred and TriPepSVM on the three independent test datasets, ATH, SC, and Human, respectively. The details of the process adopted for the comparison and the results obtained are provided below:

4.4. Performance comparison with deep-RBPPred

In Deep-RBPPred software, to make predictions, users can either choose a model trained with the balanced dataset (balance model) or a model trained with the imbalanced dataset (imbalance model). In our implementation, we extract cross-validation probabilities from the base classifiers with a balanced dataset (using SMOTE), and the meta classifier is trained with the imbalanced training dataset. Table 6 shows the comparison between the proposed method, AIRBP, and an existing imbalance Deep-RBPPRed model on three independent test datasets. Table 6 shows that AIRBP achieves an improvement of 3.96 % in ACC, 4.08 % in BACC, 7.30 % in F1-score, and 10.68 % in MCC over imbalance Deep-RBPPRed on Human test set. Likewise, AIRBP achieves an improvement of 1.39 % in ACC, 1.09 % in BACC, 1.59 % in F1-score, and 2.58 % in MCC over imbalance Deep-RBPPRed on S. *cerevisiae* test set. Similarly, AIRBP achieves an improvement of 3.05 % in ACC, 2.07 % in BACC, 2.30 % in F1-score, and 7.04 % in MCC on A. *thaliana* dataset. On the average percentage improvement over all the independent test sets, AIRBP attains an improvement of 2.80 % in ACC, 2.41 % in BACC, 3.73 % in F1-score, and 6.77 % in MCC over imbalance Deep-RBPPRed.

Table 7 shows the comparison between the proposed method and the balance Deep-RBPPRed model on three independent test datasets. Table 7 shows that AIRBP achieves an improvement of 5.15 % in ACC, 2.01 % in BACC, 7.68 % in F1-score, and 10.26 % in MCC over balance Deep-RBPPRed on Human test set. Likewise, AIRBP achieves an improvement of 14.06 % in ACC, 11.60 % in BACC, 14.29 % in F1-score, and 28.66 % in MCC over balance Deep-RBPPRed on S. *cerevisiae* test set. Moreover, on A. *thaliana* dataset, AIRBP achieves an improvement of 1.50 % in ACC, 4.73 % in BACC, 0.76 % in F1-score, and 6.60 % in MCC over balance Deep-RBPPRed. On the average percentage improvement

Table 5
Comparison of AIRBP with the existing method on training dataset through 10-fold CV.

Methods	Evaluation Metrics								
	SN (%)	SP (%)	BACC (%)	ACC (%)	FPR	FNR	PR (%)	F1-score	MCC
Deep-RBPPred Imbalance	–	–	–	–	–	–	–	–	0.730
Deep-RBPPred Balance	–	–	–	–	–	–	–	–	0.740
AIRBP	**92.02**	**97.41**	94.71	**95.84**	0.026	0.080	**93.61**	**0.928**	**0.899**
(% imp. over Deep-RBPPred Imbalance)	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(23.15 %)
(% imp. over Deep-RBPPred Balance)	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(21.49 %)

The best score values are **bold**faced. '-' represents missing value, or the value not reported by Deep-RBPPred, and '(-)' denotes that the % imp. cannot be calculated.

Table 6
Comparison between AIRBP and Imbalance Deep-RBPPred on three independent test datasets.

Methods	Dataset	Evaluation Metrics								
		SN (%)	SP (%)	BACC (%)	ACC (%)	FPR	FNR	PR (%)	F1-score	MCC
Deep-RBPPred Imbalance Model	Human	90.20	90.97	90.58	90.77	0.090	0.098	77.97	0.836	0.777
	S. cerevisiae	100.00	84.00	92.00	90.00	0.160	0.000	78.95	0.882	0.814
	A. thaliana	87.16	90.00	88.58	87.92	0.100	0.128	95.96	0.913	0.724
AIRBP	Human	**94.12**	**94.44**	**94.28**	**94.36**	**0.056**	**0.059**	**85.71**	**0.897**	**0.860**
	(% imp.)	(4.35 %)	(3.81 %)	(4.08 %)	(3.96 %)	(60.71 %)	(66.10 %)	(9.93 %)	(7.30 %)	(10.68 %)
	S. cerevisiae	**100.00**	**86.00**	**93.00**	**91.25**	**0.140**	**0.000**	**81.08**	**0.896**	**0.835**
	(% imp.)	(0.00 %)	(2.38 %)	(1.09 %)	(1.39 %)	(14.29 %)	(-)	(2.70 %)	(1.59 %)	(2.58 %)
	A. thaliana	**90.83**	**90.00**	**90.41**	**90.60**	**0.100**	**0.092**	**96.12**	**0.934**	**0.775**
	(% imp.)	(4.21 %)	(0.00 %)	(2.07 %)	(3.05 %)	(0.00 %)	(39.13 %)	(0.17 %)	(2.30 %)	(7.04 %)
	(avg. % imp.)	(2.85 %)	(2.07 %)	(2.41 %)	(2.80 %)	(25.00 %)	(-)	(4.26 %)	(3.73 %)	(6.77 %)

The best score values are **boldfaced.** Here, 'imp.' stands for improvement. The '% imp.' represents the improvement in percentage achieved by AIRBP for corresponding independent test set for corresponding evaluation metric over the imbalance Deep-RBPPRed method. Likewise, the 'avg. % imp.' represents the average percentage improvement achieved by AIRBP for all independent test set for corresponding evaluation metrics over the imbalance Deep-RBPPRed method. Additionally, '(-)' denotes that the % imp. or avg. % imp. cannot be calculated.

over all the independent test sets, AIRBP attains a gain of 6.90 % in ACC, 6.11 % in BACC, 7.57 % in F1-score, and 15.17 % in MCC over balance Deep-RBPPRed. Tables 6 and 7 show that the AIRBP outperforms the state-of-the-art Deep-RBPPRed method in both balance and imbalance models.

The best score values are **boldfaced.** Here, 'imp.' stands for improvement. The '% imp.' represents the improvement in percentage achieved by AIRBP for corresponding independent test set for corresponding evaluation metric over the balance Deep-RBPPRed method. Likewise, the 'avg. % imp.' represents the average percentage improvement achieved by AIRBP for all independent test set for corresponding evaluation metrics over the balance Deep-RBPPRed method. Additionally, '(-)' denotes that the % imp. or avg. % imp. cannot be calculated.

4.5. Comparison with TriPepSVM

Likewise, to compare AIRBP with TriPepSVM, we ran TriPepSVM on three independent test datasets. While running TriPepSVM, we discovered that it requires a Uniprot taxon id, which is by default set to 9606 (for humans). This indicates that TriPepSVM must have been trained based on the species-wise dataset. We ran TriPepSVM on the three test dataset with human taxon id, and it performs very poorly on ATH and SC datasets. So, one of the limitations of TriPepSVM is that it does not apply to the datasets of new species. The performance of TriPepSVM using Human taxon id is shown in Table 8.

From Table 8, we can conclude that TriPepSVM is not a generic method that can be applied to any species. Instead, it is strongly dependent on the Uniprot taxon id and will only perform well for particular species but not for any species. Therefore, we would like to highlight that the comparison between AIRBP and TriPepSVM is not an apple-to-apple comparison. From Table 8, AIRBP shows a very consistent performance on all the test datasets compared to the TriPepSVM method.

4.6. Time and memory costs of AIRBP

AIRBP was implemented in Python (Scikit-learn libraries). We ran the experiments on a Linux server, which consists of 64 processors and 128 GB of RAM. All 64 processors were utilized for training the proposed framework using a 10-fold cross-validation approach in parallel. Though it took approximately fifty-six minutes to train the AIRBP model, three independent test set predictions were quick and took approximately five minutes.

To profile the memory requirement of AIRBP, we have used a python memory profiler. Fig. 5. shows the memory usage of the AIRBP for each second. It is evident from Fig. 5. that AIRBP requires 20,000 MiB or 20.97 Gigabytes of memory for the model training. Further, from Fig. 5., it is also apparent that the initial high memory requirement is due to the base classifiers' cross-validation step as the base classifiers utilize individual copies of the same dataset to perform training in parallel. For meta classifier, the memory requirement reduces significantly.

The above comparison of results indicates that the proposed method, AIRBP outperforms the existing methods and is a very promising predictor. We believe that this comprehensive investigation of the ensemble-based machine learning framework and features in predicting RNA binding proteins might be useful for future proteomics studies.

5. Conclusions

In this work, we constructed an ensemble-based machine learning framework, called AIRBP, for the prediction of RNA-binding proteins (RBPs) directly from the protein sequence. The existing experimental methods for determining RBPs for millions of new proteins are not practical due to the vast amount of possible interactions to be tested. Thus, it is highly desirable to have a computational tool to prioritize the

Table 7
Comparison between AIRBP and Balance Deep-RBPPred on three independent test datasets.

Methods	Dataset	Evaluation Metrics								
		SN (%)	SP (%)	BACC (%)	ACC (%)	FPR	FNR	PR (%)	F1-score	MCC
Deep-RBPPred Balance Model	Human	98.04	86.81	92.42	89.74	0.132	0.020	72.46	0.833	0.780
	S. cerevisiae	96.67	70.00	83.33	80.00	0.300	0.033	65.91	0.784	0.649
	A. thaliana	92.66	80.00	86.33	89.26	0.200	0.073	92.66	0.927	0.727
AIRBP	Human	94.12	**94.44**	**94.28**	**94.36**	**0.056**	0.059	**85.71**	**0.897**	**0.860**
	(% imp.)	-(4.00%)	(8.79 %)	(2.01 %)	(5.15 %)	(135.71 %)	-(66.10%)	(18.29 %)	(7.68 %)	(10.26 %)
	S. cerevisiae	**100.00**	**86.00**	**93.00**	**91.25**	**0.140**	**0.000**	**81.08**	**0.896**	**0.835**
	(% imp.)	(3.44 %)	(22.86 %)	(11.60 %)	(14.06 %)	(114.29 %)	(-)	(23.02 %)	(14.29 %)	(28.66 %)
	A. thaliana	90.83	**90.00**	**90.41**	**90.60**	**0.100**	0.092	**96.12**	**0.934**	**0.775**
	(% imp.)	-(1.97 %)	(12.50 %)	(4.73 %)	(1.50 %)	(100.00 %)	-(20.65 %)	(3.73 %)	(0.76 %)	(6.60 %)
	(avg. % imp.)	-(0.84 %)	(14.72 %)	(6.11 %)	(6.90 %)	(116.67 %)	(-)	(15.01 %)	(7.57 %)	(15.17 %)

Table 8
Comparison between AIRBP and TriPepSVM on three independent test datasets with Human Taxon ID.

Methods	Dataset	Evaluation Metrics								
		SN (%)	SP (%)	BACC (%)	ACC (%)	FPR	FNR	PR (%)	F1-score	MCC
TripepSVM	Human	96.08	90.97	93.53	92.31	0.090	0.039	79.03	0.867	0.822
	S. cerevisiae	50.00	88.00	69.00	73.75	0.120	0.500	71.43	0.588	0.418
	A. thaliana	38.53	82.50	60.52	50.34	0.175	0.615	85.71	0.532	0.198
AIRBP	Human	94.12	**94.44**	**94.28**	**94.36**	**0.056**	0.059	**85.71**	**0.897**	**0.860**
	(% imp.)	-(2.04 %)	(3.81 %)	(0.80 %)	(2.22 %)	(60.71 %)	-(33.90 %)	(8.45 %)	(3.46 %)	(4.62 %)
	S. cerevisiae	**100.00**	86.00	**93.00**	**91.25**	0.140	**0.000**	**81.08**	**0.896**	**0.835**
	(% imp.)	(100.00 %)	-(2.27 %)	(34.78 %)	(23.73 %)	-(14.29 %)	(-)	(13.51 %)	(52.38 %)	(99.76 %)
	A. thaliana	**90.83**	**90.00**	**90.41**	**90.60**	**0.100**	**0.092**	**96.12**	**0.934**	**0.775**
	(% imp.)	(135.74 %)	(9.09 %)	(49.39 %)	(79.98 %)	(75.00 %)	(568.48 %)	(12.15 %)	(75.56 %)	(291.41 %)
	(avg. % imp.)	(77.90 %)	(3.54 %)	(28.32 %)	(35.31 %)	(40.48 %)	(-)	(11.37 %)	(43.80 %)	(131.93 %)

The best score values are **boldfaced.** Here, 'imp.' stands for improvement. The '% imp.' represents the improvement in percentage achieved by AIRBP for corresponding independent test set for corresponding evaluation metric over the TriPepSVM method. Likewise, the 'avg. % imp.' represents the average percentage improvement achieved by AIRBP for all independent test set for the corresponding evaluation metric over the TriPepSVM method. Additionally, '(-)' denotes that the % imp. or avg. % imp. cannot be calculated.

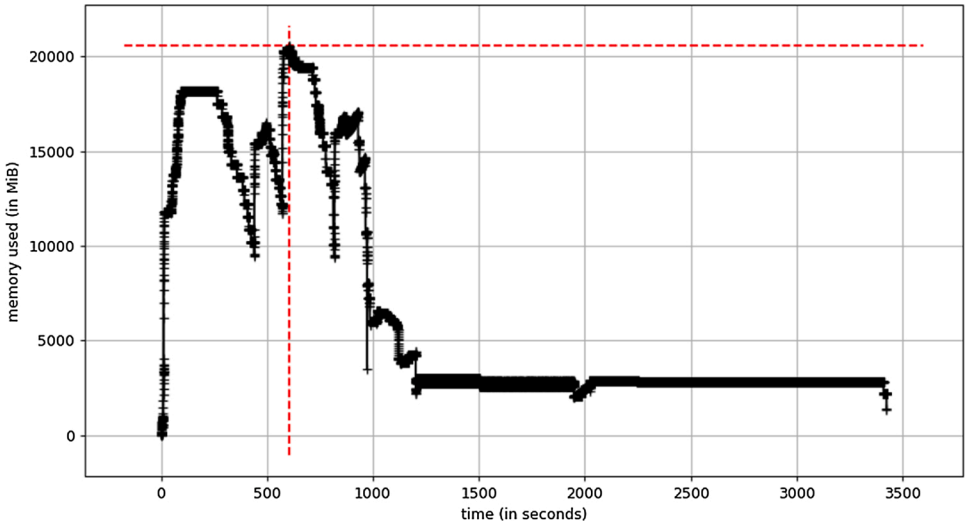

Fig. 5. Memory requirements of the AIRBP.

study of new RBPs, which is tremendously important in treating various existing and emerging critical diseases such as cancer. Towards this, the proposed RBPs predictor, called AIRBP, can be utilized by the scientific community for accurate identification and annotation of new RBPs directly from the sequence. The experimental scientist can further study these newly identified RBPs to extract valuable insight into their specific biological roles and functions.

To improve RNA-binding proteins' prediction accuracy, we have investigated and used various feature extraction and encoding techniques along with an advanced machine learning technique called stacking. We extracted multiple features, including evolutionary information, physiochemical properties, and disordered properties, and applied different encoding techniques such as composition, transition and distribution, conjoint triad, PSSM distance transformation, and residue-wise contact energy matrix transformation to encode the protein sequence in terms of features. Next, the extracted features are used to train the ensemble of predictors at the first-level (i.e., base-layer) of the stacking method. Then, the prediction probabilities from the first-level predictors are combined and used to train the predictor at the second-level (i.e., meta-layer) of the stacking method. Finally, the majority vote from RBPPred, imbalance Deep-RBPPred, and the stacking model is considered for the prediction for RBPs. The proposed ensemble framework achieves a 10-fold CV accuracy, balanced Accuracy, F1-score, and MCC of 95.84 %, 94.71 %, 0.928, and 0.899, respectively, on the training dataset. While performing the independent test, AIRBP achieves an accuracy, balanced Accuracy, F1-score, and MCC of 94.36 %, 94.28 %, 0.897, and 0.860, for the Human test set; 91.25 %, 93.00 %, 0.896, and 0.835 for S. *cerevisiae* test set; and 90.60 %, 90.41 %, 0.934 and 0.775 for A. *thaliana* test set, respectively. These promising results indicate that the ensemble framework helps improve the accuracy significantly by reducing the generalization error. Furthermore, compared to the existing better-performing method, Deep-RBPPred, the proposed AIRBP method achieves 23.15 % and 21.49 % improvement in terms of MCC based on the imbalanced and balanced training dataset, respectively. Moreover, the average percentage improvement, calculated over three different independent test sets, AIRBP outperforms imbalance Deep-RBPPred by 2.80 %, 2.41 %, 3.73 %, and 6.77 % in terms of accuracy, balanced accuracy, F1-score, and MCC, respectively. Similarly, AIRBP also outperforms the balance Deep-RBPPred model by 6.90 %, 6.11 %, 7.57 %, and 15.17 % in terms of accuracy, balanced accuracy, F1-score, and MCC, respectively.

These outcomes help us summarize that the AIRBP can be effectively used for accurate and fast identification and annotation of RNA-binding proteins directly from the protein sequence and can provide valuable insights for treating acute diseases.

Author contributions

Conceived and designed the experiments: AM, RK, MWUK, MTH. Performed the experiments: AM, RK, MWUK. Analyzed the data: AM,

RK, MWUK. Contributed reagents/materials/analysis tools: MTH. Wrote the paper: AM, RK, MWUK, MTH.

Acknowledgments

The authors gratefully acknowledge the Louisiana Board of Regents through the Board of Regents Support Fund LEQSF (2016-19)-RD-B-07.

References

[1] Beckmann BM, Horos R, Fischer B, Castello A, Eichelbaum K, Alleaume A-M, et al. The RNA-binding proteomes from yeast to man harbour conserved enigmRBPs. Nat Commun 2015;6.
[2] Anderson JSJ, Parker R. Computational Identification of cis-acting elements affecting post-transcriptional control of gene expression in Saccharomyces cerevisiae. Nucleic Acids Res 2000;28(7):1604–17.
[3] Abdelmohsen K, Kuwano Y, Kim HH, Gorospe M. Posttranscriptional gene regulation by RNA-binding proteins during oxidative stress: implications for cellular senescence. Biol Chem 2008;389(3):243–55.
[4] Qiu Y, Jiang H, Ching W-K, Ng MK. On predicting epithelial mesenchymal transition by integrating RNA-binding proteins and correlation data via L1/2-regularization method. Artif Intell Med 2019;95:96–103. 2019/04/01/.
[5] Saunus JM, French JD, Edwards SL, Beveridge DJ, Hatchell EC, Wagner SA, et al. Posttranscriptional regulation of the breast cancer susceptibility gene BRCA1 by the RNA binding protein HuR. Cancer Res 2008;68(22):9469–78.
[6] Noller HF. RNA structure: reading the ribosome. Science 2005;309(5740):1508–14.
[7] Delgado FM, Gómez-Vela F. Computational methods for Gene Regulatory Networks reconstruction and analysis: a review. Artif Intell Med 2019;95:133–45. 2019/04/01/.
[8] Baltz AG, Munschauer M, Schwanhäusser B, Vasile A, Murakawa Y, Schueler M, et al. The mRNA-Bound proteome and its global occupancy profile on protein-coding transcripts. Mol Cell 2012;46(5):674–90.
[9] Wurth L. Versatility of RNA-binding proteins in cancer. Int J Genomics 2012;2012:178525.
[10] Wang Z-L, Li B, Luo Y-X, Lin Q, Liu S-R, Zhang X-Q, et al. Comprehensive genomic characterization of RNA-Binding proteins across human cancers. Cell Rep 2018;22(1):286–98. 2018/01//.
[11] Gebauer F, Schwarzl T, Valcárcel J, Hentze MW. RNA-binding proteins in human genetic disease. Nat Rev Genet 2020. 2020/11/24.
[12] Castello A, Fischer B, Eichelbaum K, Horos R, Beckmann BM, Strein C, et al. Insights into RNA Biology from an Atlas of Mammalian mRNA-Binding Proteins. Cell 2012;149(6):1393–406.
[13] Greenberg JR. Ultraviolet light-induced cross-linking of mRNA to proteins. Nucleic Acids Res 1979;6(2):715–32.
[14] Wagenmakers AJM, Reinders RJ, Venrooij WJV. Cross-linking of mRNA to Proteins by Irradiation of Intact Cells with Ultraviolet Light. Eur J Biochem 1980;112(2).
[15] Lindberg U, Sundquist B. Isolation of messenger ribonucleoproteins from mammalian cells. J Mol Biol 1974;86(2):451–68.
[16] Castello A, Fischer B, Eichelbaum K, Horos R, Beckmann BM, Strein C, et al. Insights into RNA biology from an atlas of mammalian mRNA-binding proteins. Cell 2012;149(6):1393–406.
[17] Kwon SC, Yi H, Eichelbaum K, Föhr S, Fischer B, You KT, et al. The RNA-binding protein repertoire of embryonic stem cells. Nat Struct Mol Biol 2013;20:1122–30.
[18] Mitchell SF, Jain S, She M, Parker R. Global analysis of Yeast mRNPs. Nat Struct Mol Biol 2013;20:127–33.
[19] Zhang Z, Boonen K, Ferrari P, Schoofs L, Janssens E, Noort V v, et al. UV cross-linked mRNA-binding proteins captured from leaf mesophyll protoplasts. Plant Methods 2016;12.
[20] Marondedze C, Thomas L, Serrano NL, Lilley KS, Gehring C. The RNA-binding protein repertoire of Arabidopsis thaliana. Sci Rep 2016;6.
[21] Marondedze C, Thomas L, Gehring C, Lilley KS. Changes in the Arabidopsis RNA-binding proteome reveal novel stress response mechanisms. BMC Plant Biol 2019;19(1).
[22] Reichel M, Liao Y, Rettel M, Ragan C, Evers M, Alleaume A-M, et al. In planta determination of the mRNA-binding proteome of Arabidopsis etiolated seedlings. Plant Cell 2016;28(10):2435–52.
[23] Bach-Pages M, Homma F, Kourelis J, Kaschani F, Mohammed S, Kaiser M, et al. Discovering the RNA-binding proteome of plant leaves with an improved RNA interactome capture method. Biomolecules 2020;10(4).
[24] Si J, Cui J, Cheng J, Wu R. Computational prediction of RNA-binding proteins and binding sites. Int J Mol Sci 2015;16:26303–17.
[25] Wu CH, Apweiler R, Bairoch A, Suzek BE. The Universal Protein Resource (UniProt): an expanding universe of protein information. Nucleic Acids Res 2006;34(February).
[26] Zhao H, Yang Y, Zhou Y. Highly accurate and high-resolution function prediction of RNA binding proteins by fold recognition and binding affinity prediction. RNA Biol 2011;8(6):988–96.
[27] Zhao H, Yang Y, Zhou Y. Structure-based prediction of RNA-binding domains and RNA-binding sites and application to structural genomics targets. Nucleic Acids Res 2011;39(8):3017–25.
[28] Yang Y, Zhan J, Zhao H, Zhou Y. A new size-independent score for pairwise protein structure alignment and its application to structure classification and nucleic-acid binding prediction. Proteins 2012;80(8):2080–8.
[29] Shazman S, Mandel-Gutfreund Y. Classifying RNA-binding proteins based on electrostatic properties. PLoS Comput Biol 2008;4:e1000146.
[30] Kumar M, Gromiha MM, Raghava GPS. SVM based prediction of RNA-binding proteins using binding residues and evolutionary information. J Mol Recognit 2011;24(2):303–13.
[31] Paz I, Kligun E, Bengad B, Mandel-Gutfreund Y. BindUP: a web server for non-homology-based prediction of DNA and RNA binding proteins. Nucleic Acids Res 2016;44(W1):W568–74.
[32] Ma X, Guo J, Sun X. Sequence-based prediction of RNA-binding proteins using random forest with minimum redundancy maximum relevance feature selection. Biomed Res Int 2015;425810.
[33] Ma X, Guo J, Xiao K, Sun X. PRBP: prediction of RNA-binding proteins using a random forest algorithm combined with an RNA-binding residue predictor. IEEE/ACM Trans Comput Biol Bioinform 2015;12(6):1385–93.
[34] Zhang X, Liu S. RBPPred: predicting RNA-binding proteins from sequence using SVM. Bioinformatics 2017;33(6):854–62.
[35] Zheng J, Zhang X, Zhao X, Tong X, Hong X, Xie J, et al. Deep-RBPPred: predicting RNA binding proteins in the proteome scale based on deep learning. Sci Rep 2018;8(1).
[36] Wang Y, Chen X, Liu Z-P, Huang Q, Wang Y, Xu D, et al. De novo prediction of RNA–protein interactions from sequence information. Mol Biosyst 2013;9:133–42.
[37] Mishra A, Pokhrel P, Hoque MT. StackDPPred: a stacking based prediction of DNA-binding protein from sequence. Bioinformatics 2018;35(3):433–41.
[38] Xu R, Zhou J, Wang H, He Y, Wang X, Liu B. Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation. BMC Syst Biol 2015;9.
[39] Wolpert DH. Stacked generalization. Neural Netw 1992;5(2):241–59.
[40] Peng CR, Liu L, Niu B, Lv YL, Li MJ, Yuan YL, et al. Prediction of RNA-binding proteins by voting systems. J Biomed Biotechnol 2011;2011. pp. 506205-506205.
[41] Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 2006;22(13):1658–9.
[42] Marondedze C. The increasing diversity and complexity of the RNA-binding protein repertoire in plants. Proc R Soc B: Biol Sci 2020;287(1935):20201397. 2020/09/30.
[43] Chawla NV BK, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 2002;16:341–78.
[44] Xie S, Girshick R, Dollar P, Tu Z, He K. Aggregated residual transformations for deep neural networks. 10.1109/CVPR.2017.634.. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 5987–95.
[45] Dubchak I, Muchnik I, Holbrook SR, Kim S-H. Prediction of protein folding class using global description of amino acid sequence. Proc Natl Acad Sci U S A 1995;92:8700–4.
[46] Magnan CN, Baldi P. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. Bioinformatics 2014;30(18):2592–7.
[47] Sharma R, Raicar G, Tsunoda T, Patil A, Sharma A. OPAL: prediction of MoRF regions in intrinsically disordered protein sequences. Bioinformatics 2018;34(11):1850–8.
[48] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol 1990;215:403–10. 15 May 1990.
[49] Dubchak I, Muchnik I, Holbrook SR, Kim SH. Prediction of protein folding class using global description of amino acid sequence. Proc Natl Acad Sci U S A 1995;92(19):8700–4.
[50] Han LY, Cai CZ, Lo SL, Chung MCM, Chen YZ. Prediction of RNA-binding proteins from primary sequence by a support vector machine approach. RNA 2004;10:355–68.
[51] Zhang X, Liu S. RBPPred: predicting RNA-binding proteins from sequence using SVM. Bioinformatics 2017;33(6):854–62.
[52] Dubchak I, Muchnik I, Mayor C, Dralyuk I, Kim SH. Recognition of a protein fold in the context of the SCOP classification. Proteins 1999;35:401–7.
[53] Shen J, Zhang J, Luo X, Zhu W, Yu K, Chen K, et al. Predicting protein–protein interactions based only on sequences information. Proc Natl Acad Sci U S A 2007;104(11):4337–41.
[54] Kumar M, Gromiha MM, Raghava GP. Identification of DNA-binding proteins using support vector machines and evolutionary profiles. BMC Bioinformatics 2007;8(1):1471–2105.
[55] Kumar M, Gromiha MM, Raghava GPS. Prediction of RNA binding sites in a protein using SVM and PSSM profile. Proteins 2008;71:189–94.
[56] Kumar M, Gromiha MM, Raghava GPS. SVM based prediction of RNA-binding proteins using binding residues and evolutionary information. J Mol Recognit 2011;24(2):303–13.
[57] Mishra A, Pokhrel P, Hoque MT. StackDPPred: a stacking based prediction of DNA-binding protein from sequence. Bioinformatics 2018;bty653.
[58] Iqbal S, Mishra A, Hoque T. Improved prediction of accessible surface area results in efficient energy function application. J Theor Biol 2015;380:380–91.
[59] Zhang L, Zhao X, Kong L. Predict protein structural class for low-similarity sequences by evolutionary difference information into the general form of Chou's pseudo amino acid composition. J Theor Biol 2014;355:105–10.
[60] Calabretta S, Richard S. Emerging roles of disordered sequences in RNA-binding proteins. Trends Biol Sci 2015;40(11):662–72.
[61] Järvelin AI, Noerenberg M, Davis I, Castello A. The new (dis)order in RNA regulation. Cell Commun Signal 2016;14(9).

[62] Mishra A, Iqbal S, Hoque MT. Discriminate protein decoys from native by using a scoring function based on ubiquitous Phi and Psi angles computed for all atom. J Theor Biol 2016;398:112–21.
[63] Hoque MT, Yang Y, Mishra A, Zhou Y. sDFIRE: sequence-specific statistical energy function for protein structure prediction by decoy selections. J Comput Chem 2016;37(12):1119–24.
[64] Mishra A, Hoque MT. Three-dimensional ideal gas reference state based energy function. Curr Bioinform 2017;12(2):171–80.
[65] Zhou H, Skolnick J. GOAP: A Generalized Orientation-Dependent, All-Atom Statistical Potential for Protein Structure Prediction. Biophys J 2011;101(October): 2043–52.
[66] Babu MM, Lee Rvd, Groot NSd, Gsponer J. Intrinsically disordered proteins: regulation and disease. Curr Opin Struct Biol 2011;21(3):432–40.
[67] Dosztányi Z, Csizmók V, Tompa P, Simon I. The pairwise energy content estimated from amino acid composition discriminates between folded and intrinsically unstructured proteins. J Mol Biol 2005;347(4):827–39.
[68] Vacic V, Oldfield CJ, Mohan A, Radivojac P, Cortese MS, Uversky VN, et al. Characterization of molecular recognition features, MoRFs, and their binding partners. J Proteome Res 2007;6:2351–66.
[69] Bah A, Forman-Kay JD. Modulation of intrinsically disordered protein function by post-translational modifications. J Biol Chem 2016;291:6696–705.
[70] Lina Y-H, Qiua D-C, Changa W-H, Yehc Y-Q, Jeng U-S, Liue F-T, et al. The intrinsically disordered N-terminal domain of galectin-3 dynamically mediates multisite self-association of the protein through fuzzy interactions. J Biol Chem 2017;292:17845–56.
[71] Mohan A, Oldfield CJ, Radivojac P, Vacic V, Cortese MS, Dunker AK, et al. Analysis of Molecular Recognition Features (MoRFs). J Mol Biol 2006;362(5):1043–59.
[72] Sharma R, Sharma A, Raicar G, Tsunoda T, Patil A. OPAL+: length-specific MoRF prediction in intrinsically disordered protein sequences. Proteomics 2018; 1800058.
[73] Sharma R, Bayarjargal M, Tsunoda T, Patil A, Sharma A. MoRFPred-plus: computational identification of MoRFs in protein sequences using physicochemical properties and HMM profiles. J Theor Biol 2018;437:9–16.
[74] Mohan A, Oldfield CJ, Radivojac P, Vacic V, Cortese MS, Dunker AK, et al. Analysis of Molecular Recognition Features (MoRFs). J Mol Biol 2006;362:1043–59.
[75] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning. 2 ed. New York: Springer-Verlag; 2009.
[76] Hu Q, Merchante C, Stepanova AN, Alonso JM, Heber S. A stacking-based approach to identify translated upstream open reading frames in Arabidopsis Thaliana. International symposium on bioinformatics research and applications. 2015. p. 138–49.
[77] Iqbal S, Hoque MT. PBRpredict-Suite: a suite of models to predict peptide-recognition domain residues from protein sequence. Bioinformatics 2018. pp. bty352-bty352.
[78] Nagi S, Bhattacharyya DK. Classification of microarray cancer data using ensemble approach. Netw Model Anal Health Inform Bioinform 2013;2(3):159–73.
[79] Džeroski S, Ženko B. Is combining classifiers with stacking better than selecting the best one? Mach Learn 2004;54(3):255–73.
[80] Ho TK. Random decision forests," in document analysis and recognition, 1995. Proceedings of the Third International Conference on, Montreal, Que., Canada 1995:278–82.
[81] Breiman L. Bagging predictors. Mach Learn 1996;24(2):123–40.
[82] Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Mach Learn 2006;63 (1):3–42.
[83] 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, KDD' 16; 2016. p. 785–94.
[84] Szilágyi A, Skolnick J. Efficient prediction of nucleic acid binding function from low-resolution protein structures. J Mol Biol 2006;358(3):922–33.
[85] Altman NS. An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 1992;46:175–85.
[86] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y. LightGBM: a highly efficient gradient boosting decision tree. In: Proceedings of the 31st International Conference on Neural Information Processing Systems; 2017. p. 3149–57.
[87] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res 2012;13(February).
[88] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in python. J Mach Learn Res 2012;12.
[89] Bressin A, Schulte-Sasse R, Figini D, Urdaneta EC, Beckmann BM, Marsico A. TriPepSVM: de novo prediction of RNA-binding proteins based on short amino acid motifs. Nucleic Acids Res 2019;47(9):4406–17.