Spring 2019

Identification of RNA Binding Proteins and RNA Binding Residues Using Effective Machine Learning Techniques

Reecha Khanal
University of New Orleans

Identification of RNA Binding Proteins and RNA Binding Residues Using Effective Machine Learning Techniques

An Honors Thesis

Presented to

the Department of Computer Science

of the University of New Orleans

In Partial Fulfillment

of the Requirements for the Degree of

Bachelor of Science, with University High Honors

and Honors in Computer Science

by

Reecha Khanal

April 2019

Acknowledgements

Initially, I would like to thank Dr. Md Tamjidul Hoque and Mr. Avdesh Mishra for their continuous guidance and mentorship through my thesis and research works. I am also grateful to the Department of Computer Science at the University of New Orleans for all the resources that had been provided to me, without which this research work would not have been possible.

Furthermore, I highly appreciate the support from *Tolmas Scholars* program without which I would not be able to be involved in on-campus research activities. I would also like to thank Dr. Christopher M. Summa for being the second reader for my thesis and Ms. Erin Spence Sutherland for her continuous guidance and support through the entire tenure of my undergraduate degree.

Table of Contents

List of Tables

List of Figures

Abstract:

Identification and annotation of RNA Binding Proteins (RBPs) and RNA Binding residues from sequence information alone is one of the most challenging problems in computational biology. RBPs play crucial roles in several fundamental biological functions including transcriptional regulation of RNAs and RNA metabolism splicing. Existing experimental techniques are time-consuming and costly. Thus, efficient computational identification of RBPs directly from the sequence can be useful to annotate RBP and assist the experimental design. Here, we introduce AIRBP, a computational sequence-based method, which utilizes features extracted from evolutionary information, physiochemical properties, and disordered properties to train a machine learning method designed using stacking, an advanced machine learning technique, for effective prediction of RBPs. Furthermore, it makes use of efficient machine learning algorithms like Support Vector Machine, Logistic Regression, K-Nearest Neighbor and XGBoost (Extreme Gradient Boosting Algorithm). In this research work, we also propose another predictor for efficient annotation of RBP residues. This RBP residue predictor also uses stacking and evolutionary algorithms for efficient annotation of RBPs and RNA Binding residue. The RNA-binding residue predictor also utilizes various evolutionary, physicochemical and disordered properties to train a robust model. This thesis presents a possible solution to the RBP and RNA binding residue prediction problem through two independent predictors, both of which outperform existing state-of-the-art approaches.

Keywords: Machine Learning, Bioinformatics, RNA-Binding Proteins, RNA-Binding Residue.

Introduction:

1.1 Thesis Overview:

Today, there has been a lot of development in genomics and hence there is an increased number of proteomic data available in different online databases. Experimental methods alone are time consuming and costly. So, bioinformatics offers a faster, cheaper way to mine, evaluate and interpreted such biological data. Today, bioinformatics has become essential in dealing with biological data because of its efficiency and success in various research works. The development of computational tools for analysis and interpretation of such data through bioinformatics involves few steps: *i)* Data mining, collection, and preparation of data, *ii)* Computing to extract useful information or characteristics, can also be thought of as features, from the data, *iii)* Apply various Machine Learning Algorithms to develop a robust classifier that uses the features extracted in the previous step, and *iv)* Analyze, compare and evaluate obtained results from the classifiers. These three steps have been utilized in this thesis to develop predictors for annotation of RNA Binding Proteins and RNA Binding residues.

1.2 Contribution of the Thesis:

This thesis aims to solve one of the most important problems in bioinformatics by providing predictors for efficient annotation of both RNA Binding Proteins and RNA Binding Residues using the sequence information of the protein alone. The predictors developed could also be used to assist experimental inquiries and can also be used as a stepping stone for other prediction methods.

1.3 Technical results of the Thesis:

Technical results of this thesis can be divided into two parts: *i)* A predictor for prediction of RNA Binding Proteins, named AIRBP, and *ii)* A predictor for prediction of RNA Binding Residues. The two results are described below:

- A predictor for prediction of RNA Binding Proteins (AIRBP)

Here, we present a predictor for effective annotation of RNA Binding Proteins called AIRBP. AIRBP is a stacking based predictor that utilizes a pool of base learners like Extremely Randomized Trees, Random Forest, Logistic Regression, K-Nearest Neighbor and XGBoost (Extreme Gradient Boosting Algorithm). This predictor is fast and efficient and outperforms all other existing predictors for RNA Binding Proteins. It also provides a balanced performance on all the performance metrics and provides biologically relevant prediction of RNA Binding Proteins.

- A predictor for prediction of RNA Binding Residues

In addition to the RNA Binding Protein predictor, this research also presents a predictor for RNA Binding Residues or sites present in the RNA Binding Proteins. This predictor uses software like DisPredict, SPIDER, and SCRATCH to obtain various evolutionary, physicochemical and disordered properties of RNA Binding Proteins. This work, similar to AIRBP, which uses advanced machine learning frameworks like Stacking and Genetic Algorithms. The results from our study show that the generated predictor is well balanced on all the performance metrics and provides biologically relevant prediction of RNA Binding Residues.

1.4 Thesis Organization:

The primary aim of the thesis is to develop a computational approach for the prediction of RNA-binding proteins using only sequence information. The rest of the thesis is organized as follows: Chapter 2 discusses the design and development of RBP predictor. The details on datasets, feature extraction, and performance evaluation are provided in Chapter 2. Chapter 3 discusses the design and development of RBP residues predictor. The details on datasets, feature extraction, and performance evaluation are provided in Chapter 3. Finally, Chapter 4, concludes this thesis and states the major contributions and provides future directions and possibilities for further research to make the tools as accurate as possible.

1.5 Related Publications:

Below listed are research works that have provided noteworthy results in the world of RBP prediction.

1. Zhang, X. and Liu, S. RBPPred: predicting RNA-binding proteins from sequence using SVM. *Bioinformatics* 2017;33(6):854-862.
2. Avdesh Mishra, Reecha Khanal[§], Md Tamjidul Hoque*, "Accurate Identification of RNA-binding Proteins (AIRBP) Using Machine Learning Techniques", ***The 7th Annual Conference on Computational Biology and Bioinformatics***, Louisiana, USA, 2019 [Poster].
3. Su, H., *et al.* (2019) Improving the prediction of protein–nucleic acids binding residues via multiple sequence profiles and the consensus of complementary methods, *Bioinformatics*, **35**, 930-936.

Part of this thesis has also been presented as posters and oral presentation in the 6th and 7th Annual LA conference on Computational Biology and Bioinformatics in 2018 and 2019, respectively.

Chapter 2

AIRBP: Accurate Identification of RNA-Binding Proteins Using Machine Learning Techniques

2.1 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 (Beckmann, et al., 2015). Additionally, they contribute to several important biological functions that include RNA transport, cellular localization, gene expression, expression of histone genes, post-transcriptional gene regulation, and regulation of translation and transcription control (Glisovic, et al., 2008). As an illustration, the newly formed messenger RNA, that carries necessary genetic information from DNA to ribosomes, associates with various RNA binding proteins (RBP) to form messenger ribonucleoprotein (mRNP) complexes (Baltz, et al., 2012). These mRNP complexes govern major elements of metabolism and functions of mRNA. Similarly, the microRNPs (miRNPs), formed through association of the RBPs with microRNAs (miRNAs) controls the translation and stability of RNA itself (Wurth, 2012). The identification of RBPs along with their mRNA targets is shown useful in cancer therapy (Wurth, 2012). There are numerous other diseases that have been linked to defective RBP expression and functions, including neuropathies, muscular atrophies, and metabolic disorders (Castello, et al., 2012), highlighting the urgency of identifying possible RBPs.

As of today, numerous studies have been performed and various experimental and computational methods have been developed to identify and expand our knowledge of RBPs. 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 (Greenberg, 1979), from UV-irradiated intact cells (Wagenmakers, et al., 1980) and from untreated cells (Lindberg and Sundquist, 1974) revealed the association of a specific set of proteins with mRNA (Baltz, et al., 2012). Recently, cutting-edge experimental approaches are developed to recognize numerous RBPs, which include identification of 860 RBPs in human HeLa cells (Castello, et al., 2012) using UV crosslinking methods, 797 RBPs in human embryonic kidney cell line (Baltz, et al., 2012) using photoreactive nucleotide-enhanced UV crosslinking and oligo(dT) purification approach, 555 mRNA-binding proteins from mouse embryonic stem cells (Kwon, et al., 2013) using UV crosslinking, oligo(dT) and Mass Spectrometry and 120 RBPs from S. *cerevisiae* cells (Mitchell, et al., 2013) using UV crosslinking and purification methods. These experiments for identifying and analyzing of 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 (Si, et al., 2015). Moreover, the tremendous progress in genome sequencing has resulted in an unprecedented amount of genetic information and provided a plethora of protein sequences (Wu, et al., 2006), 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 RNA-binding preference of the target sequence (Yang, et al., 2012; Zhao, et al., 2011; Zhao, et al., 2011). Unlike template based methods, in machine learning methods the predictive model is created to make the prediction by finding a pattern in the input feature space (Kumar, et al., 2011; Paz, et al., 2016; Shazman and Mandel-Gutfreund, 2008). 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 (Zhao, et al., 2011) is a structure based approach and SPOT-seq (Zhao, et al., 2011) 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-seq achieved the Matthew's Correlational Coefficient (MCC), which is a performance evaluation parameter used in machine learning as a measure of the quality of binary classifications, of 0.57 on the benchmark 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 is used to predict RBPs. As shown, SPOT-seq achieved the MCC of 0.62 on the benchmark data of 215 RBP chains and 5765 non-binding protein chains.

The machine learning based approach for the prediction of RNA-binding proteins involves two important 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 protein (Paz, et al., 2016; Shazman and Mandel-Gutfreund, 2008); and *ii)* extraction of relevant features from protein sequence (Kumar, et al., 2011; Ma, et al., 2015; Ma, et al., 2015; Zhang and Liu, 2017). BindUp (Paz, et al., 2016) available as a web server, is one of the recent

structure-based methods that extracts electrostatic features and other properties from the structure of the protein and uses SVM classifier for RBPs prediction. As reported, BindUp attains sensitivity, a measure of proportion of actual positives that are correctly identified by a machine learning model, of 0.71 and specificity, a measure of proportion of actual negatives that are correctly identified as such by a machine learning model, 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 sequence-based approaches, Ma *et al.* (Ma, et al., 2015; Ma, et al., 2015) recently proposed two different methods, which differ in the features used to train the random forest model for predicting. In (Ma, et al., 2015), 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 (Ma, et al., 2015), the authors employed features such as a conjoint triad, binding propensity, non-binding propensity, and EIPP to establish random forest based predictor 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. Most recently, Zhang and Liu (Zhang and Liu, 2017) 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. 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 techniques and advanced machine

learning techniques that can help further improve the accuracy of RBPs predictors 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 the prediction accuracy of RNA-binding proteins and our understanding of the binding mechanism of RNA-protein interactions. 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) (Zhang and Liu, 2017), Conjoint Triad (CT) (Wang, et al., 2013; Zhang and Liu, 2017), PSSM Distance Transformation (PSSM-DT) (Mishra, et al., 2018; Xu, et al., 2015) and Residue-wise Contact Energy Matrix Transformation (RCEM-T) (Mishra, et al., 2018). Furthermore, AIRBP utilizes an ensemble machine learning framework, known as stacking (Wolpert, 1992) to predict RBPs from protein sequence only. AIRBP offers a significant improvement in the prediction of RBPs based on the benchmark and independent test datasets when compared to the existing start-of-the-art predictors. 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 stacking based machine learning technique, encoding techniques and features discussed in this work could be applied to tackle other relevant biological problems.

2.2 Methods

In this section, we describe the approach for benchmark and independent test data preparation, feature extraction and encoding, performance evaluation metrics and finally, the approach we took to establish the machine learning framework for RBPs prediction.

2.2.1 Dataset

2.2.1.1 Benchmark Dataset

For this work, we collected the updated version of the benchmark dataset from (Liu; Zhang and Liu, 2017). The updated benchmark dataset was created by the authors (Zhang and Liu, 2017) from the original benchmark dataset by removing 16 proteins that had RNAs in their crystal structure from the negative set. Therefore, the updated benchmark dataset we collected consist of 7077 non-RBPs (16 proteins removed from the original benchmark dataset which contained 7093 non-RBPs) and 2780 RBPs (same as the original benchmark dataset). Next, we found that 13 out of 2780 and 90 out of 7077 protein sequences in RBPs and non-RBPs set respectively, contained non-standard amino acids (amino acids other than the 20 standard amino acids). These sequences containing non-standard amino acids were removed from further consideration as the physiochemical properties of non-standard amino acids could not be obtained. Finally, the benchmark dataset which contains 2767 RBPs and 6987 non-RBPs was obtained and used for validation and model creation of AIRBP.

2.2.1.2 Independent Test Set

To test the performance of AIRBP, we collected the updated independent test dataset from (Liu; Zhang and Liu, 2017). This dataset consists of independent test sets for 3 species, human, S*accharomyces cerevisiae* (S. *cerevisiae*) and A*rabidopsis thaliana* (A. *thaliana*). The updated independent test set was created by the authors (Zhang and Liu, 2017) 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. The updated independent test sets contained a total of 967 RBPs and 588 non-RBPs for human, 354 RBPs and

135 non-RBPs for S. c*erevisiae* and 456 RBPs and 37 non-RBPs for A. *thaliana*. Next, we removed the protein sequences containing non-standard amino acid from each of these independent set and finally 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*.

2.2.2 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). Total of 10 different properties was encoded with a vector of 2603 dimension to represent a protein sequence as shown in Supplementary Fig. 1S. Out of 10, five distinct properties hydrophobicity, polarity, normalized van der Waals volume, polarizability and predicted secondary structure were each encoded via 21 dimension vector utilizing C-T-D encoding technique (Dubchak, et al., 1995; Zhang and Liu, 2017). 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, the properties solvent accessibility, charge, and polarity of the side chain, RCEM, and PSSM profile were encoded utilizing C-T-D, CT (Wang, et al., 2013; Zhang and Liu, 2017), RCEM transformation (Mishra, et al., 2018) and PSSM-DT transformation techniques (Mishra, et al., 2018; Xu, et al., 2015). Each of the 10 properties along with their encoding mechanism is described next in detail.

2.2.2.1 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.

2.2.2.1.1 Composition, Transition and Distribution (C-T-D) Transformation Features

The aim of C-T-D transformation method is to describe the distribution patterns of amino acid properties. This method to compute distribution patterns of amino acid properties were first suggested by (Dubchak, et al., 1995) 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 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, transition of one amino acid to other as we go linearly through the sequence, and distribution referring to how one amino acid group is distributed throughout the protein sequence (Han, et al., 2004; Zhang and Liu, 2017). To create a consistent number of features for proteins with different sequence length, 20 standard amino acids are divided into 3 groups (Dubchak, et al., 1999) based on their hydrophobicity, normalized van der Waals volume, polarity, and polarizability. Fig 1. provides an illustration of 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 technique shown in Fig.1 with an exception that the 20 standard amino acids are divided into 3 groups, 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 solvent accessibility as features, we first used the SSpro and ACCpro program (Magnan and Baldi, 2014) 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 (Magnan and Baldi, 2014) 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 feature vectors of 21 and 13 dimensions for predicted secondary structure and solvent accessibility, respectively.

Hypothetical protein sequence: **CMCAGKGKKAAACCMCMKMAGKKGHM**

Group X: C, A, G XZXXXZXZZXXXXXZXZZZXXZZXZZ

Group Z: M, K, H

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
X	Z	X	X	X	Z	X	Z	Z	X	X	X	X	X	Z	X	Z	Z	Z	X	X	Z	Z	X	Z	Z

Number of X's = 14

Number of Z's = 12

Composition:

Composition of X = (14/26) *100 = 53.846

Composition of Z = (12/26) *100 = 46.154

Transition:

Transition from X to Z or Z to X = (12/25) *100 = 48

Distribution:

	Distribution of X	**Distribution of Z**
First occurrence index	(1/26) *100 =3.846	(2/26) * 100 = 7.692
25th percentile occurrence index	(4/26) *100 =15.385	(8/26) *100 = 30.76
50th percentile occurrence index	(11/26) *100 =42.308	(17/26) *100 = 65.385
75th percentile occurrence index	(14/26) *100 =53.846	(22/26) *100 = 84.615
100th percentile occurrence index	(24/26) *100=92.308	(26/26) *100 = 100

C-T-D Output Vector:

[53.846, 46.154, 48, 3.846, 15.585, 42.308, 53.846, 92.308, 7.692, 30.769, 65.385, 84.615, 100]

Fig. 1. Illustration of C-T-D transformation technique, while 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.

2.2.2.1.2 Conjoint Triad (CT) Transformation Features

The CT transformation technique was first proposed by Shen *et al.* for protein-protein interaction prediction (Shen, et al., 2007) and has been successfully applied for protein-RNA interaction prediction in the past (Wang, et al., 2013; Zhang and Liu, 2017). In our implementation, we adopted the CT transformation technique to encode the protein sequence based on the charge and polarity of the side chain 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. Through this process, we obtain a feature vector of 64 dimensions for charge and polarity of side chains of amino acids in a protein. Supplementary Fig. 2 provides an illustration of CT transformation technique we used to extract features from protein sequence based on charge and polarity of side chains.

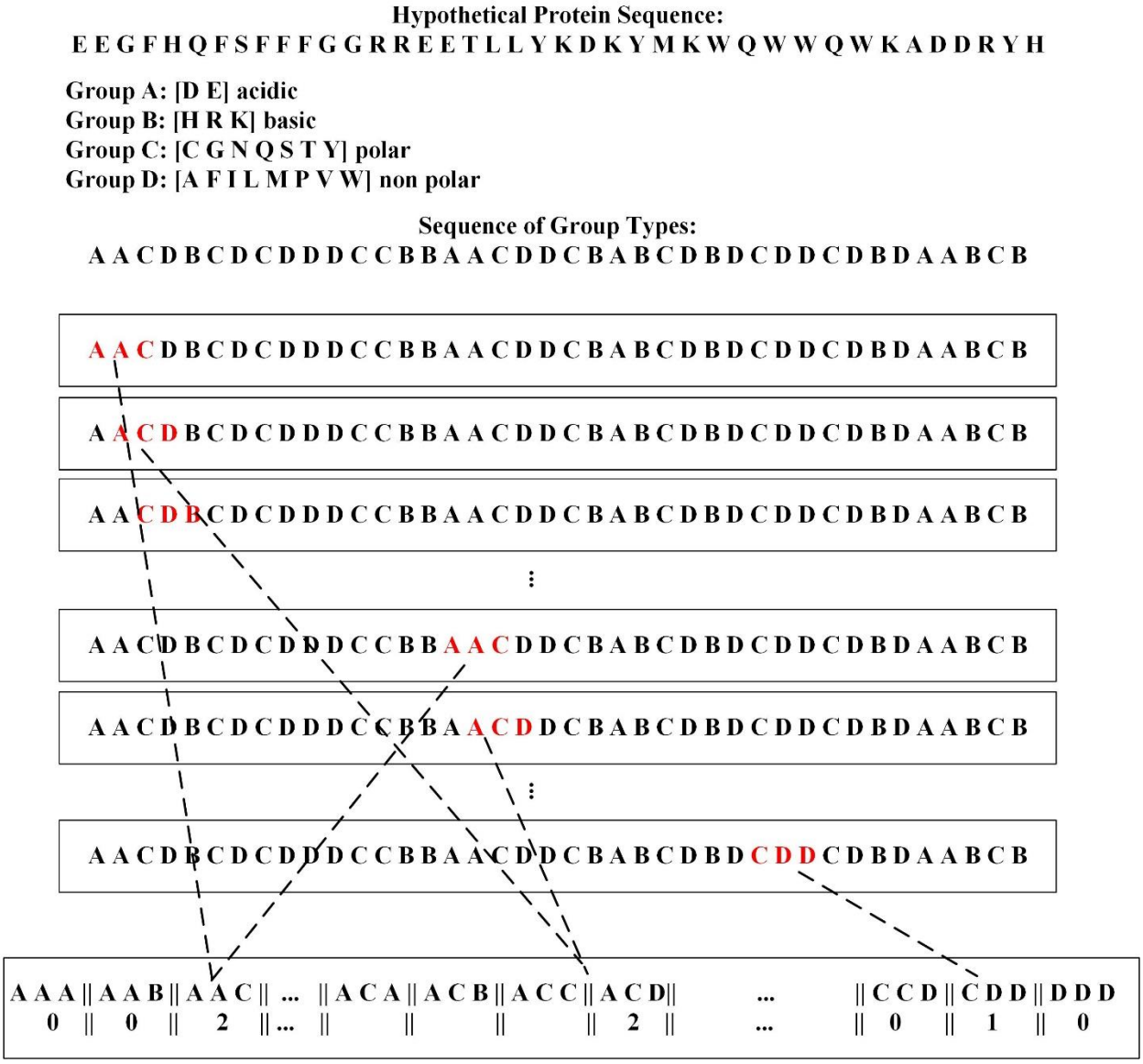

Fig. 2. Illustration of Conjoint Triad transformation technique while, the 20 standard amino acids are divided into 4 groups (Group A, B, C and D representing acidic, basic, polar and non-polar, respectively).

2.2.2.2 Features Extracted from Evolutionary Information

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

Evolutionary information is one of the most important information useful for solving various biological problems and has been widely used in many research work (Iqbal, et al., 2015; Kumar, et al., 2007; Kumar, et al., 2008; Kumar, et al., 2011; Mishra, et al., 2018; Zhang and Liu, 2017). In this work, the evolutionary information in the form of 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. High score in the PSSM matrix indicates more conserved positions and the lower score indicates less conserved positions (Mishra, et al., 2018). 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 (Altschul, et al., 1990). The evolutionary information in PSSM profile is represented as a matrix of L*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.

2.2.2.2.1 PSSM-Distance Transformation (PSSM-DT) Features

We use two types of distance transformation techniques (Mishra, et al., 2018; Xu, et al., 2015): *i)* the PSSM distance transformation for same pairs of amino acids (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:

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

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 × *K* number of features were generated where *K* is the maximum range of *D* (*D* = 1,2, …, *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:

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

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

2.2.2.2.2 Evolutionary Distance Transformation (EDT) Features

Unlike PSSM-DT, the EDT approximately measures the non-co-occurrence probability of two amino acids separated by a certain distance *d* in a protein sequence from the PSSM profile (Mishra, et al., 2018; Zhang, et al., 2014). The EDT is calculated from the PSSM profile as:

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

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 $D = L_{min}-1$ where, L_{min} is the length of the shortest protein sequence in the benchmark dataset. Using EDT, we obtain a feature vector of dimension 400.

2.2.2.3 Features Extracted to Account for Intrinsically Disordered Regions

In this section we describe a feature extraction technique utilized to obtain a fixed dimensional feature vector from residue-wise contact energy matrix, to encode protein sequence.

RBPs are found to bind with RNA not only through classical structured

RNA-binding domains but also through intrinsically disordered regions (IDRs) (Calabretta and Richard, 2015). For example, approximately 20% of the identified mammalian RBPs (~170 proteins) were found to be disordered by over 80% (Järvelin, et al., 2016). 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 (Hoque, et al., 2016; Iqbal, et al., 2015; Mishra and Hoque, 2017; Mishra, et al., 2016; Zhou and Skolnick, 2011) as IDPs lack a defined and ordered 3D structure (Babu, et al., 2011). Therefore, to inherently incorporate important information regarding the IDRs and amino acid interactions, we employed the predicted residue-wise contact energies (Dosztányi, et al., 2005) and molecular recognition features (MoRFs) (Sharma, et al., 2018), to encode the protein sequence.

2.2.2.3.1 Residue-Wise Contact Energy Matrix Transformation (RCEM-T) Features

We adopted the predicted residue-wise contact energy matrix (RCEM) derived in (Dosztányi, et al., 2005), 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 which contains residue-wise contact energy for 20 standard amino acids. For a protein sequence of length *L*, an *L* × 20 dimensional matrix *M* is obtained which holds 20 dimensional vector for each amino acid in a protein sequence. The resulting matrix *M* is then encoded into a feature vector of 20 dimensions by computing the column-wise sum as:

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

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 element in the final *RCEM-T* vector can be represented as:

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

Table 1. RCEM table 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	0.2	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

2.2.2.3.2 Molecular Recognition Features (MoRFs)

MoRFs, also sometimes known as molecular recognition elements (MoREs), are disordered regions in a protein those exhibit various molecular recognition and binding functions (Vacic, et al., 2007). 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 (Bah and Forman-Kay, 2016; Lina, et al., 2017). MoRFs play a vital role in various biological functions of IDPs located within long disordered protein sequences (Mohan, et al., 2006; Sharma, et al., 2018; Sharma, et al., 2018; Sharma, et al., 2018). Additionally, Mohan *et al.* suggest that functionally significant residual structures exist in MoRF regions prior to the actual binding (Mohan, et al., 2006). These residual structures could, therefore, be useful in the prediction of binding between proteins and RNA. Here, to capture functional properties of IDRs which 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 (Sharma, et al., 2018). Then, 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.

2.3 Performance Evaluation

To evaluate the performance of AIRBP, we adopted a widely used 10-fold cross-validation (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 same size. When one fold is kept aside for testing, the remaining 9 folds are used to train the classifier. This process of training and test is repeated until each fold has been kept aside once for testing and consequently, the test accuracies of each fold are combined to compute the average (Hastie, et al., 2009). Unlike a 10-fold CV, in independent testing, the classifier is trained with the benchmark dataset and consequently tested using the independent test dataset. While independent testing, it is ensured that none of the samples in the independent test set are present in the benchmark dataset. We used several performance evaluation metrics listed in

Table 2 as well as ROC and AUC to test the performance of the proposed method as well as to 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 proteins).

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)	$\frac{1}{2}(\frac{TP}{TP+FN}+\frac{TN}{TN+FP})$
Precision (PR)	$\frac{TP}{TP+FP}$
F1-score (Harmonic mean of precision and recall)	$\frac{2TP}{2TP+FP+FN}$
Mathews Correlation Coefficient (MCC)	$\frac{(TP*TN)-(FP*FN)}{\sqrt{(TP+FN)*(TP+FP)*(TN+FP)*(TN+FN)}}$

2.4 Feature Selection

In this section, we discuss the feature selection approaches that we adopted to select relevant features. During the feature extraction process, we collected a feature vector of 2603 dimensions, which is significantly large. Therefore, to reduce the feature space and select the relevant features that could help improve the classification accuracy, we adopted two distinct feature selection approaches, namely Incremental Feature Selection (IFS) and Genetic Algorithm (GA) based feature selection.

Feature Selection using IFS

IFS starts with an empty feature vector and a feature group is added to the feature vector if the addition of the feature group to the feature vector improves the performance of the predictor. In case, by adding the new feature group, the accuracy of the predictor is reduced, this feature group is discarded, and a new feature group is tested in an iterative fashion. During IFS, we performed 10-fold CV on benchmark dataset using XGBoost as a predictor. The values of XGBoost parameters: max_depth, eta, silent, objective, num_class, n_estimators, min_child_weight, subsample, scale_pos_weight, tree_method and max_bin were set to 6, 0.1, 1, 'multi:softprob', 2, 100, 5, 0.9, 3, 'hist' and 500, respectively and the rest of the parameters were set to their default value. We used ACC as the evaluation metric to decide whether the new feature group will be added to the feature vector or not. In our implementation of IFS, only Vander Waals Volume feature group was ignored from the feature vector as the addition of this feature decreased the ACC of the predictor. Therefore, through IFS, 2582 features out of 2603 features were selected as relevant features.

Feature Selection using GA

GA is a population-based stochastic search technique that mimics the natural process of evolution. It contains a population of chromosomes where each chromosome represents a possible solution to the problem under consideration. In general, a GA operates by initializing the population randomly, and by iteratively updating the population through various operators including elitism, crossover and mutation to discover, prioritize and recombine good building blocks present in parent chromosomes to finally obtain fitter chromosome (Hoque, et al., 2010; Hoque, et al., 2007; Hoque and Iqbal, 2017).

Encoding the solution of the problem under consideration in the form of chromosomes and computing the fitness of the chromosomes are two important steps in setting up the GA. Here, to perform feature selection, we encode each feature f_i in our feature space $F = [f_1, f_2, \cdots, f_n]$ by a single bit of 1/0 in a chromosome space where, the value of 1 represents that the *i*-th feature is selected and the value of 0 represents that the *i*-th feature is not selected. The length of the chromosome space is equal to the length of the feature space. Moreover, to compute the fitness of the chromosome, we use the XGBoost algorithm (Chen and Guestrin, 2016). The choice of XGBoost was made because of its fast execution time and reasonable performance compared to other machine learning classifiers. During feature selection, the values of XGBoost parameters: max_depth, eta, silent, objective, num_class, n_estimators, min_child_weight, subsample, scale_pos_weight, tree_method, and max_bin were set to 6, 0.1, 1, 'multi:softprob', 2, 100, 5, 0.9, 3, 'hist' and 500, respectively and the rest of the parameters were set to their default value. In our implementation, the objective fitness is defined as:

$$obj\_fit = ACC + AUC + MCC \quad (6)$$

where, ACC is the accuracy, AUC is the area under the receiver operating characteristic curve and MCC is the Matthews Correlation Coefficient. To evaluate the fitness of the chromosome, a new data space *D* is obtained which only includes the features for which the chromosome bit is 1. The values of ACC, AUC and MCC metrics of the *obj_fit* are obtained by performing 10-fold CV on a new data space *D* using the XGBoost algorithm. Furthermore, the additional parameters of the GA in our implementation were set to a population size of 20, maximum generation to 300, elite-rate to 5%, crossover-rate to 90% and mutation rate to 50%. Through this GA based feature selection, only 1346 features out of 2603 features were selected as relevant features. Therefore, we were able to achieve two-fold benefits from the GA based features selection which are significantly reduced feature space and relevant features. Finally, we noticed that at least one of the features from each type of features we extracted was present in the feature set selected by GA. Therefore, all the feature types extracted in this study were found to be important for the prediction of RBPs.

2.5 Framework of AIRBP

To develop the AIRBP predictor for RBPs prediction, we adopted the idea of a stacking based machine learning approach (Wolpert, 1992) which, has recently been successfully applied to solve various bioinformatics problems (Hu, et al., 2015; Iqbal and Hoque, 2018; Mishra, et al., 2018; Nagi and Bhattacharyya, 2013). 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. Stacking framework 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 are employed (Džeroski and Ženko, 2004). 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 principles are different from one another (Mishra, et al., 2018; Nagi and Bhattacharyya, 2013).

In order to select the classifiers to use in the first and second level of the AIRBP stacking framework, we analyzed the performance of six individual classification methods: *i)* Random Decision Forest (RDF) (Ho, 1995); *ii)* Bagging (Bag) (Breiman, 1996); *iii)* Extra Tree (ET) (Geurts, et al., 2006); iv) Extreme Gradient Boosting (XGBoost or XGB) (Chen and Guestrin, 2016); v) Logistic Regression (LogReg) (Hastie, et al., 2009; Szilágyi and Skolnick, 2006); and vi) K-Nearest Neighbor (KNN) (Altman, 1992).

All the classification methods mentioned above are built and optimized using python's Scikit-learn library (Pedregosa, et al., 2012). In order to design stacking framework for AIRBP, we evaluated the different combination of base-classifiers and finally selected the one that provided the highest performance. The set of stacking framework tested are:

i) SF1: RDF, XGBoost, LogReg, KNN in base-level and XGBoost in meta-level,

ii) SF2: Bag, XGBoost, LogReg, KNN in base-level and XGBoost in meta-level and

iii) SF3: ET, XGBoost, 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 (Mishra, et al., 2018). 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 combination SF1, SF2 and SF3, the XGBoost classifier is used both in the base as well as in the meta-level because it performed best among all the other individual methods applied in this work. While examining the 10-fold CVs performance of the above three combinations, we found that the first stacking framework, SF1 attains the highest performance. Therefore, we employ four classifiers RDF, XGBoost, LogReg, and KNN as the base classifiers and another XGBoost as the meta-classifier in AIRBP stacking framework. In AIRBP, the probabilities of both the classes (RBP and non-RBP) generated by the four base-classifiers are combined with the 1346 features selected by GA and provided as an input features to the meta-classifier which eventually provides the prediction for RBPs. Fig. 3 illustrates the prediction framework of the AIRBP.

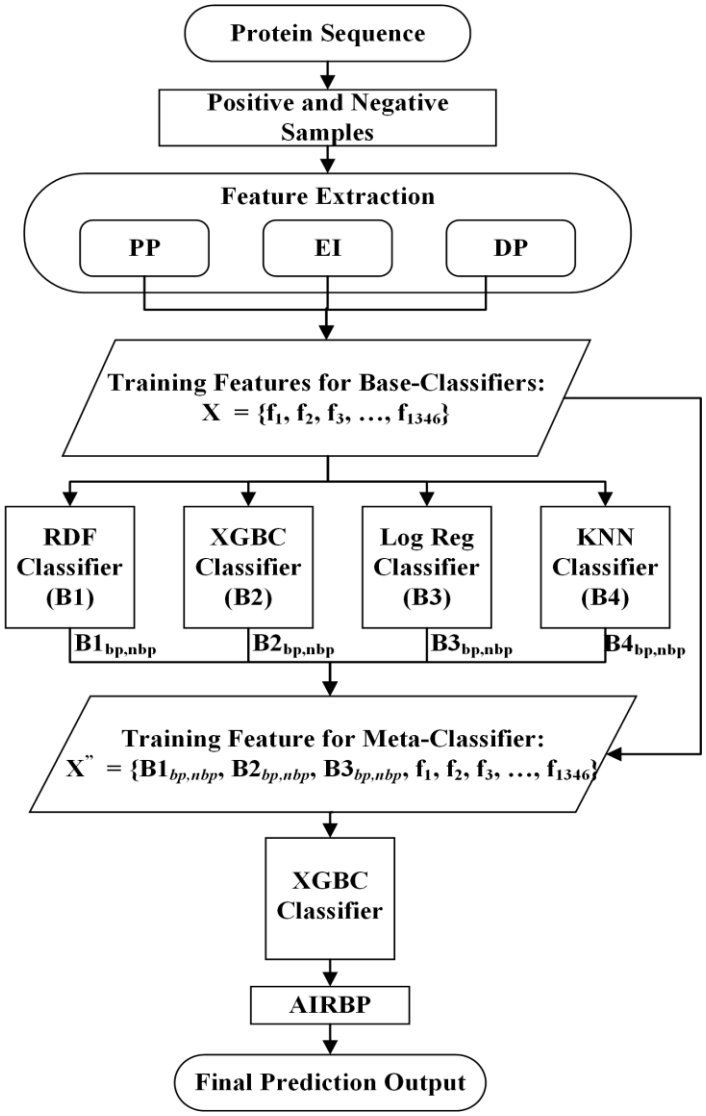

Fig 3. Illustration of the AIRBP framework.

2.6 Results

In this section, we first demonstrate the results of the feature selection. Then, we show the performance comparison of potential base-classifiers and stacking frameworks. Finally, we report the performance of AIRBP on the benchmark dataset and three independent test datasets and consequently compare it with the existing method.

2.6.1 Feature Selection

To reduce the feature space and select the relevant features that support the classification accuracy, we adopted the IFS and GA based feature selection approach. Through IFS and GA, 2582 and 1346 features out of 2603 total features were selected as relevant features, respectively. From Table 3, we observe that IFS could not reduce the feature space as significantly as GA. Additionally, the performance of XGBoost after IFS did not improve significantly and is lower than the performance resulted by the GA-based feature selection. We found that the benefit of GA feature selection were two folds, significant reduction of feature space and identification of relevant features along with improved performance. In Supplementary Table 3S, we show the performance comparison of XGBoost based predictor before and after IFS and GA-based feature selection.

Table 3. Comparison of RBPs prediction results on benchmark dataset before and after feature selection.

Algorithm	Num. of Features	Evaluation Metrics								
		SN (%)	SP (%)	BACC (%)	ACC (%)	FPR	FNR	PR (%)	F1-score	MCC
XGBoost Before Feature Selection	2603	82.11	96.81	89.46	92.64	0.03	0.18	91.06	0.86	0.82
XGBoost After IFS	2582	82.26	96.92	89.59	92.76	0.03	0.18	91.37	0.87	0.82
XGBoost After GA-based Feature Selection	1346	**89.13**	**96.95**	**91.03**	**93.59**	**0.03**	**0.15**	**91.71**	**0.88**	**0.84**

Best scores are **bold** faced.

2.6.2 Selection of Classifiers for Stacking

To select the methods to use as the base and the meta-classifiers, we analyzed the performance of six different machine learning algorithms: RDF, Bag, ET, XGBoost, LogReg, and KNN on the benchmark dataset through 10-fold CV approach. The performance comparison of the individual classifiers on the benchmark dataset is shown in Table 4.

Table 4. Comparison of various machine learning algorithms on the benchmark dataset through 10-fold CV.

Metric/Methods	Bag	KNN	LogReg	RDF	XGBoost	ET
SN (%)	82.18	57.54	82.00	72.24	**89.09**	67.44
SP (%)	96.84	89.17	96.39	98.47	97.48	**98.58**
BACC (%)	89.51	73.35	89.20	85.36	**93.28**	83.01
ACC (%)	92.68	80.19	92.31	91.03	**95.10**	89.75
FPR	0.032	0.108	0.036	0.015	0.025	**0.014**
FNR	0.178	0.425	0.180	0.278	**0.109**	0.326
PR (%)	91.14	67.77	90.00	94.92	93.34	**94.96**
F1-score	0.866	0.622	0.858	0.820	**0.912**	0.789
MCC	0.816	0.492	0.807	0.775	**0.878**	0.742

Best score values are **bold** faced.

Table 4 further shows that the optimized XGBoost is the best performing classifier among six different classifiers implemented in our study, in terms of sensitivity, balanced accuracy, accuracy, FNR, F1-score, and MCC. Moreover, the optimized XGBoost attains sensitivity, balanced accuracy, accuracy, FNR, F1-score, and MCC of 89.09%, 93.28%, 0.109, 0.912, and 0.878, respectively. Besides, the ET classifier attains the highest specificity, FPR, and precision of 98.58%, 0.014, and 94.96%, respectively. As the benchmark dataset is highly imbalanced, we consider MCC as the deciding scores as it provides the balanced measure of any predictor trained on an imbalanced dataset. Furthermore, it is evident from Table 1 that the MCC of the optimized XGBoost is 18.33%, 13.29%, 8.79%, 78.46%, and 7.59% higher than ET, RDF, LogReg, KNN,

and Bag, respectively. The greater performance of the XGBoost algorithm motivated us to use it both as a base as well as a meta-classier in the AIRBP prediction framework.

To further select the classifiers to be used at the base-level, we adopted the guidelines of base-classifier selection based on different underlying principles. Therefore, we used KNN and LogReg as two additional classifiers at the base-level. Then, we added single tree-based ensemble method out of three methods, RDF, Bag, and ET, at a time as the fourth base-classifier and designed three different combinations of stacking framework, namely SF1, SF2, and SF3. The performance comparison of SF1, SF2 and SF3 stacking framework on the benchmark dataset using 10-fold CV are presented in Table 5.

Table 5. Comparison of different stacking framework with different set of base-classifiers on benchmark dataset through 10-fold CV.

Metric/Methods	SF1	SF2	SF3
SN (%)	90.17	89.99	**90.53**
SP (%)	**97.44**	97.15	97.29
BACC (%)	93.80	93.57	**93.91**
ACC (%)	**95.38**	95.12	95.38
FPR	0.026	**0.028**	0.027
FNR	0.098	**0.100**	0.095
PR (%)	**93.31**	92.59	92.98
F1-score	**0.917**	0.912	0.917
MCC	**0.885**	0.879	0.885

Best scores are **bold** faced.

Table 5 demonstrates that SF1, which includes RDF, XGBoost, LogReg, and KNN as base-classifiers and another XGBoost as a meta-classifier outperformed SF2 and SF3. Hence, we select SF1 as our final predictor of RBPs.

2.6.3 Performance Comparison with Existing Approaches on the Benchmark Dataset

Here, we compare the performance of AIRBP with RBPPred (Zhang and Liu, 2017) on the benchmark dataset using the 10-fold CV approach. RBPPred is a top performing existing approach for the prediction of RBPs directly from the sequence. Furthermore, it is to be noted that AIRBP uses the same benchmark dataset as RBPPred therefore, for the comparison, the quantities for all the evaluation metrics for RBPPred are obtained from Zhang and Liu (Zhang and Liu, 2017). The prediction results of AIRBP and RBPPred on benchmark dataset computed using 10-fold CV are listed in Table 6.

Table 6. Comparison of AIRBP with existing method on benchmark dataset through 10-fold CV.

Metric/Methods	RBPPred	AIRBP (% imp.)
SN (%)	83.07	**90.17** (8.55%)
SP (%)	96.00	**97.44** (1.50%)
BACC (%)	-	93.80 (-)
ACC (%)	92.36	**95.38** (3.26%)
FPR	-	0.026 (-)
FNR	-	0.098 (-)
PR (%)	89.00	**93.31** (4.84%)
F1-score	0.859	**0.917** (6.75%)
MCC	0.808	**0.885** (9.53%)

Here, best scores are **bold** faced. The '% imp.' stands for percentage improvement and '-' represents missing value or the value not reported by RBPPred and '(-)' represents that the % imp. cannot be calculated.

From Table 6, we observed that AIRBP outperforms RBPPred based on all the evaluation metrics applied in this study. Particularly, AIRBP provides 8.55%, 1.50%, 3.26%, 4.84%, 6.75% and 9.53% improvement over RBPPred based on SN, SP, ACC, PR, F1-score and MCC, respectively. In addition, in Table 3, we report the values of BACC, FPR, and FNR only for the AIRBP predictor as the values of these metrics were not reported by RBPPred. Since our benchmark dataset is highly imbalanced (contains 2767 RBPs and 6987 non-RBPs) which also reflects the natural frequency, we focus on comparing the predictors based on MCC and F1-score. MCC considers true and false

positives as well as negatives and is generally considered as a balanced measure which can be used even though the classes are of very different sizes. Likewise, F1-score is the harmonic average of the precision and recall and is generally considered another balanced measure when the dataset is imbalanced. Since F1-score considers harmonic average, it is considered to provide an appropriate score to the model rather than an arithmetic mean. From Table 3, it is clear that based on MCC and F1-score AIRBP outperforms RBPPred by 9.53% and 6.75%.

2.6.4 Performance Comparison with Existing Approaches on the Independent Test Set

In this section, we further compare the performance of AIRBP with RBPPred predictor on three different independent test sets, Human, S. cerevisiae and A. thaliana. Here, we only report the comparison of AIRBP with RBPPred because RBPPred is the top performing sequence-based predictor of RBPs in the literature. As reported, RBPPred provides much better performance than SPOT-seq (Zhao, et al., 2011) and RNApred (Kumar, et al., 2011) predictors, which are the only two additional sequence-based methods that can be accessed either through a web server or code is publicly available for download. To perform independent testing, we first train AIRBP on complete benchmark dataset and subsequently test it on three different independent test sets, Human, *S. cerevisiae* and *A. thaliana*. The predictive results of AIRBP and RBPPred on three different independent test sets are compared in Table 4. Table 4 indicates that AIRBP achieves an improvement of 9.32% in SN, 4.54% in ACC, 4.19% in F1-score and 8.50% in MCC over RBPPred on Human test set. Likewise, AIRBP achieves an improvement of 9.51% in SN, 4.41% in ACC, 3.52% in F1-score and 8.23% in MCC over RBPPred on *S. cerevisiae* test set. Furthermore, while testing on *A. thaliana*, AIRBP achieves an improvement of 6.61% in SN, 5.34% in ACC, 4.28% in PR, 3.03% in F1-score and 10.61% in MCC over RBPPred approach.

Table 7: Comparison of AIRBP with an existing method using independent test sets.

Methods	Dataset	Evaluation Metrics								
		SN (%)	SP (%)	BACC (%)	ACC (%)	FPR	FNR	PR (%)	F1-score	MCC
RBPPred	Human	84.28	96.65	-	89.00	-	-	97.65	0.905	0.788
	S. cerevisiae	86.16	94.59	-	87.73	-	-	96.52	0.910	0.729
	A. thaliana	86.40	94.59	-	87.02	-	-	94.59	0.925	0.537
AIRBP	Human (% imp.)	**92.14** (9.32%)	94.52 (-2.21%)	93.33 (-)	**93.04** (4.54%)	0.055 (-)	0.079 (-)	96.53 (-1.14%)	**0.943** (4.19%)	**0.855** (8.50%)
	S. cerevisiae (% imp.)	**94.35** (9.51%)	84.33 (-10.85%)	89.34 (-)	**91.60** (4.41%)	0.157 (-)	0.057 (-)	94.09 (-2.52%)	**0.942** (3.52%)	**0.789** (8.23%)
	A. thaliana (% imp.)	**92.11** (6.61%)	86.11 (-8.97%)	89.11 (-)	**91.67** (5.34%)	0.139 (-)	0.079 (-)	**98.82** (4.28%)	**0.953** (3.03%)	**0.594** (10.61%)
	(avg.% imp.)	(8.48%)	(-7.34%)	(-)	(4.76)	(-)	(-)	(0.21%)	(3.58%)	(9.11%)

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 RBPPred method. Likewise, the 'avg. % imp.' represents the average percentage improvement achieved by AIRBP for all independent test set for corresponding evaluation metric over the RBPPred method. Additionally, '-' represents missing value or the value not reported by RBPPred and '(-)' represents that the % imp. or avg. % imp. cannot be calculated.

Moreover, while analyzing the average percentage improvement over all the independent test sets AIRBP attains improvement of 8.48% in SN, 4.76% in ACC, 0.21% in PR, 3.58% in F1-score and 9.11% in MCC over RBPPred. Besides, RBPPred seems to be 7.34% better in an average over three test sets in terms of SP (i.e. predicting negative samples or non-RBPs) over AIRBP. However, AIRBP provides 0.21% improvement in an average over three test sets in terms of PR over RBPPred. Additionally, as stated above, for the imbalanced dataset the F1-score and MCC are widely used as a balanced measure between sensitivity and specificity. Our predictor, AIRBP shows consistent improvement in F1-score and MCC over RBPPred for all three independent test set. Specifically, AIRBP provides 4.19% and 8.05% improvement in F1-score and MCC, respectively over RBPPred while tested on Human test set. Similarly, AIRBP shows 3.52% and 8.23% improvement in F1-score and MCC, respectively over RBPPred on *S. cerevisiae* as well as 3.03% and 10.61% improvement in F1-score and MCC, respectively over RBPPred on *A. thaliana*

test set. Finally, based on an average percentage improvement (calculated over three different datasets) in F1-score and MCC, AIRBP outperforms RBPPred by 3.58% and 9.11%.

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 stacking based machine learning framework and features in predicting RNA binding proteins might be useful for future proteomics studies.

2.7 Conclusions

In this work, we constructed a stacking based machine learning framework, called AIRBP, for the prediction of RNA-binding proteins directly from the protein sequence. To improve the prediction accuracy of RNA-binding proteins, we have investigated and used various feature extraction and encoding techniques, various feature selection techniques along with an advanced machine learning technique called stacking. We extracted various features including evolutionary information, physiochemical properties, and disordered properties and applied various 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, we applied two different feature selection techniques incremental feature selection and genetic algorithm based feature selection to identify the relevant features as well as to significantly reduce the feature space. Next, only the relevant features are used to train the ensemble of predictors at the first-level (i.e. base-layer) of the stacking framework. Then, the prediction probabilities from the first-level predictors are combined with the originally selected features and used to train the predictor at the second-level (i.e. meta-layer) of the stacking framework. Finally, the AIRBP stacking framework achieves a 10-fold CV accuracy, F1-score,

and MCC of 95.38%, 0.917 and 0.885 respectively, on the benchmark dataset. While performing the independent test, AIRBP achieves an accuracy, F1-score, and MCC of 93.04%, 0.943 and 0.855, for Human test set; 91.60%, 0.942 and 0.789 for S. cerevisiae test set; and 91.67%, 0.953 and 0.594 for A. thaliana test set, respectively. These promising results indicate that the stacking framework helps improve the accuracy significantly by reducing the generalization error. Furthermore, in comparison with the top performing method, RBPPred, AIRBP achieves 3.26%, 6.75% and 9.53% improvement in terms of accuracy, F1-score and MCC respectively, based on a benchmark dataset. F1-score and MCC are two widely used measures for the imbalanced dataset. Moreover, the average percentage improvement, calculated over three different independent test sets, AIRBP outperforms RBPPred by 4.76%, 3.58% and 9.11% in terms of 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 critical diseases.

Chapter 3

Prediction of RNA Binding Residues using Advanced Machine Learning Techniques

3.1 Introduction:

RNA Binding proteins (RBPs) are essential components that play pivotal roles in several cellular and developmental processes like gene expression, gene regulation, protein synthesis, posttranscriptional splicing and cellular localization of mRNAs (Beckmann, et al., 2015). RBPs also interact with several types of RNAs, such as mRNA, tRNA, and rRNA and hence are important in several biological processes. Defects in RBPs have been also linked to critical diseases like cancer, several immunological disorders and neurodegenerative diseases in humans (Walia, et al., 2016).

Eukaryotic genomes encode many RBPs. For Example, In yeast, 5–8% of genes encode RBPs, and in *Caenorhabditis elegans* and *Drosophila melanogaster*, approximately 2% of the genome is annotated to encode RBPs (Pruitt, et al., 2014). The human genome encodes more than 1500 different RBPs (Walia, et al., 2016). Despite such abundance of RBPs in nature, we know very little about these proteins.

Several efforts have been made to understand RBPs and RNA and protein interactions in general. Some oldest and direct ways of determining and understanding complex structures of protein-RNA complexes date back to almost four decades through experiments such as X-ray and/or NMR (Su, et al., 2019). These and other several experimental methods have provided us a lot of information about nucleic acid and protein interactions that otherwise would have been unknown. Despite their great importance, the experimental methods are expensive, time-consuming and labor intensive.

Today, there is an abundance of known proteins and protein sequences due to great progress in genome sequencing. There is also a rapid accumulation of the proteins, Nucleic Acids (RNA and DNA) data. For instance, as of November 2014, NCBI's RefSeq database (Pruitt, et al., 2014) includes about 47 million nonredundant proteins and more than 9 million RNA and DNA transcripts from about 49 000 organisms[1] (Yan, et al., 2015).

Thus, with an increasing number of protein and RNA data, there is growing demand and urgent need of faster and accurate computational algorithms to gain more information about these data in a faster and automated manner. These accurate and fast computational methods could also assist the ongoing experimental techniques. Many diverse computational studies have been conducted which has focused on wide range of problems like prediction of RNA binding proteins and DNA Binding proteins (Mishra, et al., 2018; Zhang and Liu, 2017), recognition of DNA-binding domain/protein(Si, et al., 2015), DNA motif pair discovery(Wong, 2017), and more. Prediction and understanding of protein-RNA interactions would be a huge help in knowing more about these increasing amounts of uncategorized protein and RNA sequence data.

In this work, we focus on the computational prediction of RNA Binding residues (i.e., residues that directly contact RNA) from RNA-binding protein chains. The ability to computationally predict which residues of a protein directly participate in the RNA-binding process can help us understand the mechanisms of protein-RNA interactions. The computational methods for identification of RNA binding residues can be divided into two broad categories: *i)* template based; and *ii)* machine learning based. Template based methods detect significant structural or sequence similarity between the query and a template known to bind RNA, to assess the RNA-binding

[1] http://www.ncbi.nlm.nih.gov/refseq/

preference of the target sequence. Template based method is a more definitive method to identify RNA binding residues. Template based method extracts the RNA binding residue from a high-resolution experimental structure of a protein-RNA complex (Walia, et al., 2016). However, these three dimensional structures used by template based methods are available for only a small fraction of known protein-RNA complexes. Such three dimensional protein structures are very difficult to obtain. The number of solved protein-RNA complex structures in the Protein Data Bank (PDB) is only 1661 out of 114,402 total structures as of December 16, 2015 (Walia, et al., 2016). These methods can also be thought of as structure based method (Ren and Shen, 2015; Zhang, et al., 2010; Zhao, et al., 2011). They use structure derived features such as solvent-accessible surface area or secondary structure to make predictions.

Machine Learning based methods, however, learn to make predictions by finding a pattern in input feature space. Some examples of Machine Learning methods are support vector machines (SVM), neural networks, random forest, naïve Bayes classifier, nearest neighbors algorithm, and other ensemble classifiers. Machine Learning based methods can be very useful to identify RNA binding residue in the absence of a three dimensional structure. They make use of sequence information of a protein/RNA sequence which is much more easily available and is abundant nowadays. These methods can also be thought as of sequence based methods (Terribilini, et al., 2007; Wang and Brown, 2006). They use sequence derived features such as amino acid identity or physiochemical properties to make predictions. For instance, BindN is a machine learning based approach for classification of RNA Binding Residue that uses support vector machines (SVM). Results indicate that BindN achieves an accuracy of 69.32%, along with a sensitivity of 66.28% and a specificity of 69.84% (Wang and Brown, 2006). Pprint is a machine learning based method that uses SVM trained on PSSM profile generated by PSI-BLAST search of 'nr' protein database. Results indicate

that it achieves a prediction accuracy of 75.53% and *MCC* value of 0.44 while predicting RNA-interacting amino acids.

There are also few methods that combine both template based and machine learning based methods to achieve a predictor that can utilize benefits of both types of methods (Su, et al., 2019; Terribilini, et al., 2007). (Su, et al., 2019) is a method that combined both template based, and sequence based methods. It is also the best performing predictor of DNA and RNA binding residue.

In this study, we present a sequence based method for prediction of RNA Binding residues. We explore different features, encoding techniques, and machine learning approaches, to further improve the prediction accuracy and further understand the binding mechanisms of RNA-protein interactions with higher accuracy. Finally, we propose an RBP residue predictor that utilizes Evolutionary, physicochemical and disordered features of a protein. We believe that the proposed predictor can be used to predict RNA binding residues effectively from sequence information alone. Moreover, the proposed predictor can also be applied to solve other relevant biological problems.

3.2. Methods

This section describes the process of benchmark and test dataset preparation, feature extraction and encoding, performance evaluation metrics and finally, the machine learning methods and framework developed for this work.

3.2.1 Dataset

For this work, we used three benchmark datasets YFK16, YK17 and MW15 These three datasets were collected from recent studies (Miao and Westhof, 2015; Su, et al., 2019; Yan, et al., 2015;

Yan and Kurgan, 2017). These datasets contain subsets for DNA binding proteins and/or RNA binding proteins. We only extracted RNA Binding proteins from each of these datasets because our research work only focuses on RNA Binding Residues. RNA Binding protein contents in each of these datasets are described below:

YFK16: This dataset considers both 3.5 and 5 Å cutoff. The sequence identity between the training and test proteins is less than 30%. For training dataset, this database contains 158 RBPs extracted at a sequence identity cutoff of 3.5 Å and 158 RBPs in extracted at a sequence identity cutoff of 5 Å. Furthermore, for test dataset, this dataset consists of 17 RBPs extracted at a sequence identity cutoff of 3.5 Å and another 17 RBPs in extracted at a sequence identity cutoff of 5 Å (Su, et al., 2019).

YK17: This dataset is an extension of YFK16 database by the inclusion of more structures that were released after YFK16 was published. All the protein sequences in this database were extracted at a sequence identity cutoff of 3.5 Å. This dataset contains a total of 339 RBPs in training dataset and 49 RBPs in the test data set (Su, et al., 2019).

MW15: This dataset is used as independent test dataset and includes 15 RBPs extracted at a sequence identity cutoff of 5 Å. Sequence identity between this dataset and the previous two datasets is less than 25%. This dataset contains no training data (Su, et al., 2019).

3.2.2 Feature Extraction

Aiming to create an efficient RBP residue predictor using sequence information alone, a feature vector for each amino acid in a protein sequence was derived using PSSM profile, Physicochemical Properties, Residue-wise Contact Energy Matrix (RCEM) and Molecular Recognition Features (MoRFs).

3.2.2.1 Physicochemical Properties

We obtained seven features for physicochemical properties. These features were extracted directly from DisPredict2 (Iqbal and Hoque, 2015). The features we obtained, include steric parameters, hydrophobicity, volume, polarizability, isoelectric point, helix probability, and sheet probability (Meiler, et al., 2001).

3.2.2.2 Residue Wise Contact Energy Matrix (RCEM) feature

To get the RCEM feature we used the residue-wise contact energy matrix (RCEM matrix) derived in (Dosztányi, et al., 2005), by the least square fitting of 674 proteins primary sequence with the contact energies derived from the tertiary structure of 785 proteins. The RCEM matrix is a 20 × 20 dimensional matrix which contains residue-wise contact energy for 20 standard amino acids. For a protein sequence of length *L*, an *L* × 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 dimensions by computing the column-wise sum. 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. Hence, a 20 dimensional feature vector is obtained for each amino acid residue of a protein sequence.

3.2.2.3 Half Sphere Exposure (HSE) and Torsion angles

Half Sphere Exposure(HSE) was first introduced and used in (Hamelryck, 2005). HSE is a measure of amino acid residue exposure in protein. HSE can be calculated by division of contact number (CN) sphere into two halves by a plane perpendicular to the Cβ-Cα vector. Two measures produced from the division of CN sphere are called HSE-up and HSE-down. For our work, we

used SPIDER2 (Heffernan, et al., 2016; Heffernan, et al., 2018 (in press); Heffernan, et al., 2017) to extract HSE-up and HSE-down feature. Additionally, the backbone structure of a protein can be described by torsion angles Phi (φ) and Psi (ψ).

Torsion angles are the important structural descriptor for proteins. They are also important in understanding and predicting protein structure, function, and interactions. In our study, we employed predicted φ and ψ angles as features which were also extracted from SPIDER2 program.

3.2.2.4 Molecular Recognition Features (MoRFs)

MoRFs are disordered protein regions that exhibit various binding and molecular recognition functions. Post-translational modifications (PTMs) can induce disorder-to-order transitions of intrinsically disordered proteins (IDPs) upon binding to RNA, DNA or other molecules. They are also suggested to be present prior to actual binding, so can be a good indicator of binding. In this work, we obtain a single predicted MoRFs score from the OPAL program (Sharma, et al., 2018). Hence one dimensional feature vector is obtained for each amino acid residue.

3.2.2.5 Position Specific Scoring Matrix (PSSM)

PSSM captures the evolution derived information in proteins. Evolutionary information is one of the most important information useful for solving various biological analysis and is also used in many studies (Biswas, et al., 2010; Iqbal and Hoque, 2016; Iqbal and Hoque, 2018; Iqbal, et al., 2015; Islam, et al., 2016; Mishra, et al., 2018; Verma, et al., 2010). Furthermore, evolutionarily conserved residues are found to play crucial functional roles such as binding (Glaser, et al., 2003).

The normalized PSSM value was used for this software and the PSSM values were obtained using DisPredict. DisPredict2 internally executes three iterations of PSI-BLAST (Altschul, et al., 1990) against NCBI's non-redundant database to generate a PSSM profile and subsequently converts it

to normalized PSSM by dividing each value by a value of 9. PSSM is a matrix of L×20 dimensions, where L is the length of the protein. The rows in PSSM represent the position of amino acid in the sequence and the columns represent the 20 standard amino acid types. Hence, every residue in the protein sequence is encoded by a 20-dimensional feature vector.

3.2.2.6 Amino Acid Type

To obtain this feature a single numerical value out of 20 is provided. Each number representing one type of amino acid residue out of the 20 standard amino acid residues.

3.2.2.7 Ordered or Disordered Proteins

This is a single dimensional feature vector obtained from DisPredict. This feature is simply representing whether a protein residue is ordered or disordered. A value of +1 represents a disordered residue while a value of -1 represents an ordered residue.

3.2.2.8 Monogram (MG) and Bigram (BG)

Monogram and Bigram were extracted from DisPredict which internally used computed these two features using PSSM. MG and BG represent the conserved evolutionary information in three dimensional structural levels. Monogram feature matrix consists of one monogram value (MG) for each type of amino acid and bigram feature matrix consists of one bigram value (BG) for each pair of 20 possible amino acids, respectively.

3.2.2.9 Position Specific Estimated Energy (PSEE) and Terminal Indicator

Position Specific Estimated Energy (PSEE) has been found and empirically verified to effectively classify disorder versus ordered residues and can segregate different secondary structure type residues by computing the constituent energies. PSEE can also be useful in the detection of the

existence of the critical binding regions (Iqbal and Hoque, 2016). We used DisPredict to extract the value of PSEE. PSEE was indicated by a feature vector of length one.

Terminal Indicator feature helps distinguish the terminal residues for their position specific disorder like behavior. We extracted the value of the terminal indicator from DisPredict. DisPredict included terminal indicator feature (T) by encoding five residues of N-terminal as {−1.0, −0.8, −0.6, −0.4, −0.2} and C-terminal as {+1.0, +0.8, +0.6, +0.4, +0.2} respectively, whereas the rest of the residues were labeled 0.0. The terminal indicator was hence also represented by a feature vector of length one.

3.2.2.10 Backbone dihedral torsion angles (dphi and dpsi)

Many protein functions are a result of the flexible motion of the protein backbone. This backbone flexibility of a protein can be described by the fluctuation of backbone torsion angles. Dihedral torsion angles (dphi and dphi) are believed to provide a complete description of the backbone. By significant change of torsion angles of only a few amino-acid residues, it can result in many functional motions (Zhang, et al., 2010).

Two feature vectors for each amino acid representing the backbone dihedral torsion angles are extracted from DisPredict software itself. The DisPredict software runs DAVAR internally to get values for theses torsion angles.

3.2.2.11 Accessible Surface Area (ASA)

ASA is a structural feature that is found to be very effective for the prediction of binding sites. To obtain ASA probability SPIDER2 was used. SPIDER2 that utilizes three iterations of deep learning neural networks to improve the prediction accuracy of several structural properties simultaneously.

It also achieves a correlation coefficient of 0.76 between predicted and actual solvent accessible surface area (Yang, et al., 2016).

3.2.2.12 Solvent Accessibility (SA) and Secondary Structure (SS)

We use SCRATCH (Cheng, et al., 2005) to obtain Secondary Structure(SS) and Solvent Accessibility(SA) of protein residues. SCRATCH, in turn, uses SSPro (Pollastri, et al., 2002) to predict SS of a protein based on sequence homology and structure homology. SSPro gives its output in terms of three classes (helix, strand and other) to represent the secondary structure of a protein. We modified the output obtained from SCRATCH through SSPRO to a three dimensional feature vector where 1 represents presence and 0 represents the absence of helix or strand or other (Cheng, et al., 2005). For instance, a feature vector 1, 0, 0 represents the presence of helix, a feature vector of 0,1,0 represents the presence of strand and, a feature vector of 0,0,1 represents the presence of other structures.

Similarly, SCRATCH, in turn, uses ACCPro (Pollastri, et al., 2002) to predict SA of protein residues. The prediction is based on one dimensional recurrent neural network that takes PSI-BLAST generated homologs as input. Each residue in a protein is predicted as either buried or exposed residue (Cheng, et al., 2005). We modified the output for SA from SCRATCH into a one dimensional feature vector where 1 represents the exposed residue and 0 represents the buried residue.

3.2.3 Feature Selection using Genetic Algorithm

For this work during the feature selection process Genetic Algorithm is used. GA is a population-based stochastic search technique that mimics the natural process of evolution. It contains a population of chromosomes where each chromosome represents a possible solution to the problem

under consideration. In general, a GA operates by initializing the population randomly, and by iteratively updating the population through various operators including elitism, crossover and mutation to discover, prioritize and recombine good building blocks present in parent chromosomes to finally obtain fitter chromosomes (Hoque, et al., 2010; Hoque, et al., 2007; Hoque and Iqbal, 2017).

Encoding the solution of the problem under consideration in the form of chromosomes and computing the fitness of the chromosomes are two important steps in setting up the GA. Here, to perform feature selection, we encode each feature f_i in our feature space $F = [f_1, f_2, \cdots, f_n]$ by a single bit of 1/0 in a chromosome space where, the value of 1 represents that the *i*-th feature is selected and the value of 0 represents that the *i*-th feature is not selected. The length of the chromosome space is equal to the length of the feature space. Moreover, to compute the fitness of the chromosome, we use XGBoost algorithm (Chen and Guestrin, 2016). The choice of XGBoost was made because of its fast execution time and reasonable performance compared to other machine learning classifiers. During feature selection, the values of XGBoost parameters: max_depth, eta, silent, objective, num_class, n_estimators, min_child_weight, subsample, scale_pos_weight, tree_method, and max_bin were set to 6, 0.1, 1, 'multi:softprob', 2, 100, 5, 0.9, 3, 'hist' and 500, respectively and the rest of the parameters were set to their default value. In our implementation, the objective fitness is defined as:

$$obj\_fit = ACC + AUC + MCC \tag{6}$$

where, ACC is the accuracy, AUC is the area under the receiver operating characteristic curve and MCC is the Matthews Correlation Coefficient.

3.2.4 Window selection

While considering binding residues it is important to consider neighboring residues too. Including the information of neighboring residues helps to consider the effect of optimal interactions among amino acid residues due to contacts among neighboring residues. The contacts among neighboring residues have been found to play essential roles in determining the structure of proteins and the way in which protein folding occurs. An optimal size of the sliding Window (W) was searched to determine the number of residues around a target residue, which can moderate the RNA-protein interaction. We designed 20 different machine learning models using XGBoost classifier and 20 different window sizes (3, 5, 7, 9, 11, 13, 15 ,17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, and 41). Window size for which the XGBoost classifier yields the highest 10-fold cross-validation(CV) balanced accuracy and AUC was selected as the optimal window size for the classifier.

Table 8. Performance of various window sizes on the combined benchmark dataset using the XGBoost Classifier.

Window /Metric	SN	SP	BACC	ACC	FPR	FNR	Precision	F-Score	MCC
3	31.510	98.245	64.877	87.914	0.01755	0.68490	76.77	0.44665	0.44097
5	32.486	98.268	65.377	88.085	0.01732	0.67514	77.456	0.45774	0.45148
7	33.180	98.284	65.732	88.205	0.01716	0.66820	77.980	0.46552	0.45882
9	33.254	98.313	65.783	88.242	0.01687	0.66746	78.311	0.46684	0.46075
11	33.327	98.345	65.836	88.280	0.0165	0.66673	78.667	0.46819	0.46278
13	33.235	98.374	65.804	88.290	0.01626	0.66765	78.918	0.46773	0.46315
15	33.278	98.409	65.843	88.326	0.01591	0.66722	79.298	0.46882	0.46504
17	**33.677**	**98.400**	**66.038**	**88.380**	**0.01600**	**0.66323**	**79.401**	**0.47295**	**0.46845**
19	33.075	98.453	65.764	88.332	0.01547	0.66925	79.654	0.46742	0.46500

21	33.419	98.433	65.926	88.369	0.01567	0.66581	79.623	0.47079	0.46745
23	33.432	98.465	65.948	88.397	0.01535	0.66568	79.956	0.47149	0.46892
25	33.425	98.438	65.932	88.374	0.01562	0.66575	79.672	0.47093	0.46770
27	33.180	98.433	65.807	88.332	0.01567	0.66820	79.506	0.46820	0.46517
29	33.401	98.492	65.946	88.416	0.01508	0.66509	80.224	0.47165	0.46980
31	33.444	98.466	65.995	88.400	0.01534	0.66556	79.974	0.47164	0.46909
33	33.358	98.482	65.920	88.400	0.01518	0.66642	80.097	0.47100	0.46895
35	33.591	98.491	66.041	88.444	0.01509	0.66409	80.302	0.47368	0.47155
37	33.290	98.490	65.890	88.397	0.01510	0.66710	80.148	0.47041	0.46865
39	33.223	98.527	65.875	88.417	0.01473	0.66777	80.509	0.47036	0.46963
41	33.468	98.538	66.003	88.465	0.01462	0.66532	80.744	0.47322	0.47245

Selected window size is **bold** faced.

From the table above, the optimal performance is found at the window size of 17 which has Sensitivity of 33.677%, Specificity of 98.400%, Balanced Accuracy of 66.038 and Overall Accuracy of 88.380. The best Specificity and Precision is obtained at window size 41 which is 98.538 and 80.744 respectively. However, the window size of 17 was used since it provides the best balanced accuracy which is an important metric for measuring the predictive performance of various machine learning methods in an imbalanced dataset.

3.2.5 Performance Evaluation

To evaluate the performance of our predictor, we used 10-fold CV and the independent testing approach. In 10-fold CV, the dataset is divided into 10 equal(more or less) parts. When one fold is kept aside for testing, the remaining 9 folds are used to train the classifier. This process of

training and test is repeated until each fold has been kept aside once for testing and consequently, the test accuracies of each fold are combined to compute the average (Hastie, et al., 2009).

For independent testing, the classifier is trained with the benchmark dataset and consequently tested using the independent test dataset. While independent testing, it is ensured that none of the samples in the independent test set are present in the benchmark dataset. We used several performance evaluation metrics listed in Supplementary Table 1S as well as ROC and AUC to test the performance of the proposed method as well as to 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 residue).

3.2.6 Framework of our RBP residue Predictor:

To develop our predictor for efficient prediction of RNA Binding Protein residues, we have used an advanced ensemble based machine learning based approach called stacking (Wolpert, 1992). Stacking has been successfully used in many research works and has also been proved to be useful in solving various bioinformatics problems (Hu, et al., 2015; Iqbal and Hoque, 2018; Mishra, et al., 2018; Nagi and Bhattacharyya, 2013). Stacking collects information from multiple models in different phased and combines them to form a new model. Stacking is also considered to yield better results than individual methods by reduction of the generalization error.

In order to select the classifiers to use in the first and second level of the AIRBP stacking framework, we analyzed the performance of six individual classification methods: *i)* Random Decision Forest (RDF) (Ho, 1995); *ii)* Logistic Regression (LogReg); iii) Extreme Gradient Boosting (XGBoost or XGB) (Chen and Guestrin, 2016); iv) K-Nearest Neighbor (KNN) (Altman,

1992); and *v)E*xtra Tree (ET) (Geurts, et al., 2006); (Hastie, et al., 2009; Szilágyi and Skolnick, 2006). These classification methods in addition to their configuration details are briefly discussed below.

i) *RDF:* RDF (Ho, 1995) constructs a multitude of decision trees on various sub-samples of the dataset and outputs the mean prediction of the decision trees to improve the predictive accuracy and control over-fitting. In our implementation of the RDF, we used bootstrap samples to construct 1,000 trees in the forest.

ii) *KNN:* KNN (Altman, 1992) operates by learning from the *K* number of training samples closest in distance to the target point in the feature space. The classification decision is computed from the majority votes coming from the neighbors. In this work, the value of *K* was set to 9 and all the neighbors were weighted uniformly.

iii) XGB: XGB (Chen and Guestrin, 2016) follows the principle of gradient boosting. It also uses a more regularized model formalization to control over-fitting, which results in better performance. In addition to better performance, XGB is designed to provide higher computational speed. In our implementation of the XGB, we used 100 bosting stages with a softprob, a parameter for XGB, learning objective, where the number of classes was set to 2 as we are dealing with a binary classification problem of carbohydrate-binding and non-carbohydrate-binding residues. The values of the additional parameters: learning rate, maximum depth, minimum child weight, and subsample ratio were set to 0.1, 3, 5 and 0.9, respectively.

iv) *LOGREG:* We implemented LOGREG (Hastie, et al., 2009; Szilágyi and Skolnick, 2006) with *L2* regularization as another classifier to be used in staking framework. It measures the relationship between the dependent categorical variable (in our case: a carbohydrate-binding or non-carbohydrate-binding) and one or more independent variables by generating an estimation

probability using logistic regression. The parameter, *C* which controls the regularization strength is optimized to achieve the best 10-fold CV balanced accuracy using grid search (Bergstra and Bengio, 2012). In our implementation, the optimal value of the parameter, C was found to be 2.3784.

v) *ET:* We employed extremely randomized tree or ET (Geurts, et al., 2006) as another classifier to be used in stacking framework. ET fits several randomized decision trees from the data sample and uses averaging to improve the prediction accuracy and control over-fitting. We constructed the ET model with 1,000 trees and the quality of a split was assessed by the Gini impurity index. The rest of the parameters for our model were set to default.

All of the above discussed classifiers and machine learning methods were implemented using python's Scikit-learn library (Pedregosa, et al., 2012). We further evaluated three different combinations of base-classifiers and finally selected the one that provided the best performance. Three sets of machine learning frameworks tested are described below:

iv) **SF1:** ET, KNN, RDF in base-level and ET in meta-level,

v) **SF2:** ET, LogReg, RDF in base-level and ET in meta-level and

vi) **SF3:** ET, XGB, RDF in base-level and ET in meta-level.

The choice of base-level classifiers is made such that the classifiers are different from each other and the individual performance on the training dataset using 10-fold cross-validation.

3.3 Results

In this section, we demonstrate the results of the individual machine learning methods on the benchmark training dataset with 10 fold cross validation (CV). Then we show the performance and performance comparison of the stacking frameworks on the benchmark training dataset with 10 fold CV.

3.3.1 Selection of Classifiers for Stacking

To select the individual machine learning methods to be used as base and meta-classifiers, we analyzed the performance of five different machine learning algorithms: KNN, LogReg, RDF, XGB, and ET on the benchmark dataset through 10-fold CV. The performance comparison of the individual classifiers on the benchmark dataset is shown in the table below.

Table 9. Comparison of various machine learning methods on benchmark dataset with 10-fold CV.

Metric/Methods	KNN	LogReg	RDF	XGB	ET
SN (%)	21.011	35.501	82.729	33.677	**82.802**
SP (%)	96.693	97.017	99.267	98.400	**99.519**
BACC (%)	58.852	66.259	90.998	66.038	**91.16**
ACC (%)	84.977	87.494	96.707	88.380	**96.931**
FPR	0.03307	0.02983	**0.00733**	0.01600	0.00481
FNR	0.78989	0.64499	**0.17271**	0.66323	0.17198
PR (%)	53.780	68.548	95.384	79.401	**96.924**
F1-score	0.30216	46.776	88.607	47.295	**0.89308**
MCC	0.26864	43.313	86.993	46.845	**0.87899**

Best score values are **bold** faced.

The table above further shows that the ET is the best performing classifier among the five tested classifiers in terms of sensitivity, specificity, balanced accuracy, overall accuracy, precision, F1-score, and MCC. Moreover, the ET classifier attains a sensitivity, balanced accuracy, overall accuracy, precision, F1-score, and MCC of 82.802%, 99.519%, 91.16%, 96.931%, 96.924%,

0.89308, and 0.87890. Because of the greater performance of ET algorithm, we chose to use it as both meta and base classifier for our predictor.

To further select the classifiers to be used at the base-level, we adopted the guidelines of base-classifier based on different underlying principles. We created three combinations of stacking frameworks on the benchmark dataset using 10-fold CV as presented in the table below.

Table 10: Comparison of different stacking framework with a different set of base-classifiers on the benchmark dataset through 10-fold CV.

Metric/Methods	SF1	SF2	SF3
SN (%)	83.066	**83.189**	83.164
SP (%)	99.528	**99.525**	99.524
BACC (%)	91.297	**91.357**	91.344
ACC (%)	96.979	**96.996**	96.992
FPR	0.00472	0.00475	**0.00476**
FNR	**0.16934**	0.16811	0.16836
PR (%)	**96.989**	96.979	96.972
F1-score	0.89489	**0.89556**	0.89539
MCC	0.88097	**0.88167**	0.88147

Best score values are **bold** faced.

The table above demonstrates that SF2, which includes ET, Log Reg, and RDF as base-level classifiers and ET as a meta-level classifier outperformed SF1 and SF3. Hence, we select SF2 to be our final predictor.

3.3.2 Future Work

This research work can be further extended to perform test on the independent a test dataset. The performance of our predictor on the test datasets will also allow the comparison of our proposed method with other state-of-the-art methods. Future works will also focus on applying other more complex machine learning methods like support vector machine (SVM). Further, we can also focus on annotation of RNA-binding residues with a focus on complex stacking based architecture. We

can also create two separate predictors for datasets with different cutoff (3.5A and 5A). The methods will also be tested on several other datasets so as to establish consistency in performance.

3.4 Conclusions

In this research work, we have developed a stacking based machine learning predictor for prediction of RNA Binding Protein residues. We collected a benchmark dataset of three independent training data set and four independent testing data set to train and independently test our predictor. Several features were extracted and chosen with the help of the genetic algorithm to train our predictor. In addition to this advanced ensemble based machine technique, called stacking, was implemented to create a robust classifier. To utilize advantages of stacking we combined the output from the base-learners with the original features and used it as an input to the predictor at second-stage (i.e., meta-layer). These outcomes can help us conclude that our predictor can be used for efficient annotation of RNA Binding Protein Sites and could also provide insights in curing critical diseases.

Chapter 4

Conclusions and Recommendations

The aim of this work was to characterize of new biological properties of proteomic data in order to categorize the proteomic data itself. Computational modeling of various properties of RNA binding proteins was done in this research work. This work can be useful in better understanding proteins, RNAs and their interactions in nature. The comprehensive research objective addressed the applications in the following three disciplines:

1) Bioinformatics: It involves the process of development and implementation of various tools using novel algorithms to solve important biological problems.
2) Computational Biology: It involves the analysis and interpretation of the protein and amino acid data along with their structures to perform efficient analysis of the biological features
3) Machine Learning: It involves the development of novel machine learning algorithms to perform advanced analysis like pattern finding in data and get as much information from the feature space as possible.

4.1 Summary

In this research work, two predictors have been developed, namely AIRBP and RNA Binding Residue Predictor. Summary of these two predictors are listed below:

AIRBP: We have developed a predictor, AIRBP, for efficient annotation of RNA Binding Proteins. Here we perform large scale data collection from online databases, extract evolutionary, physicochemical and disordered features from the data obtained and finally apply advanced

machine learning techniques to these data in order to create a predictor that can efficiently predict RNA Binding Proteins. This predictor outperforms other state-of-the-art methods.

RNA Binding Residue Predictor: Using similar processes a predictor for RNA Binding Residue is developed in this work. This predictor utilizes features created by software like DisPredict, SPIDER and SCRATCH. This predictor is trained on evolutionary, physicochemical and disordered features, and uses advanced machine learning techniques like Genetic Algorithm and Stacking.

4.2 Future Scope

The predictors developed here, in this research work can be used to assist experimental methods that have been ongoing to better understand protein-RNA interaction. It can be very useful in categorizing a large amount of protein data that has been discovered today due to developments in genomics and proteomics. Moreover, the methods used in this research work can also be used by other predictors, methods like stacking and genetic algorithm are generic and can be used in other fields besides the study of RNAs and proteins.

5.0 References

Altman, N.S. (1992) An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression, *The American Statistician*, **46**, 175-185.

Altschul, S.F., *et al.* (1990) Basic Local Alignment Search Tool., *J. Mol. Biol.*, **215**, 403-410.

Beckmann, B.M., *et al.* (2015) The RNA-binding proteomes from yeast to man harbour conserved enigmRBPs, *Nature Communications*, **6**.

Bergstra, J. and Bengio, Y. (2012) Random Search for Hyper-Parameter Optimization, *Journal of Machine Learning Research*, **13**.

Biswas, A.K., Noman, N. and Sikder, A.R. (2010) Machine learning approach to predict protein phosphorylation sites by incorporating evolutionary information., *BMC Bioinformatics*, **11**.

Chen, T. and Guestrin, C. (2016) XGBoost: a scalable tree boosting system. In, *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, pp. 785-794.

Cheng, J., *et al.* (2005) SCRATCH: a protein structure and structural feature prediction server, *Nucleic Acids Research*, **33**, W72–W76.

Dosztányi, Z., *et al.* (2005) The pairwise energy content estimated from amino acid composition discriminates between folded and intrinsically unstructured proteins, *Journal of Molecular Biology*, **347**, 827-839.

Geurts, P., Ernst, D. and Wehenkel, L. (2006) Extremely randomized trees, *Machine Learning*, **63**, 3-42.

Glaser, F., *et al.* (2003) ConSurf: identification of functional regions in proteins by surface-mapping of phylogenetic information., *Bioinformatics*, **19**, 163-164.

Hamelryck, T. (2005) An amino acid has two sides: A new 2D measure provides a different view of solvent exposure., *Proteins: Structure, Function, Bioinformatics*, **59**, 38-48.

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning*. Springer Series in Statics. Springer-Verlag New York.

Heffernan, R., *et al.* (2016) Highly accurate sequence-based prediction of half-sphere exposures of amino acid residues in proteins., *Bioinformatics*, **32**, 843-849.

Heffernan, R., *et al.* (2018 (in press)) Single-sequence-based prediction of protein secondary structures and solvent accessibility by deep whole-sequence learning., *Journal of Computational Chemistry*.

Heffernan, R., *et al.* (2017) Capturing non-local interactions by long short term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers, and solvent accessibility., *Bioinformatics*, **33**, 2842-2849.

Ho, T.K. (1995) Random decision forests. *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*. IEEE, Montreal, Que., Canada, pp. 278-282.

Hoque, M.T., *et al.* (2010) DFS Generated Pathways in GA Crossover for Protein Structure Prediction, *Neurocomputing*, **73**, 2308-2316.

Hoque, M.T., Chetty, M. and Sattar, A. (2007) Protein Folding Prediction in 3D FCC HP Lattice Model using Genetic Algorithm. *IEEE Congress on Evolutionary Computation (CEC) Singapore*. Singapore, pp. 4138-4145.

Hoque, M.T. and Iqbal, S. (2017) Genetic algorithm-based improved sampling for protein structure prediction, *International Journal of Bio-Inspired Computation*, **9**, 129-141.

Hu, Q., *et al.* (2015) A Stacking-Based Approach to Identify Translated Upstream Open Reading Frames in Arabidopsis Thaliana. *International Symposium on Bioinformatics Research and Applications*. Bioinformatics Research and Applications, pp. 138-149.

Iqbal, S. and Hoque, M.T. (2015) DisPredict: A Predictor of Disordered Protein Using Optimized RBF Kernel, *PLOS one*, **10**.

Iqbal, S. and Hoque, M.T. (2016) Estimation of position specific energy as a feature of protein residues from sequence alone for structural classification, *PLOS ONE*, **11**, e0161452.

Iqbal, S. and Hoque, M.T. (2018) PBRpredict-Suite: a suite of models to predict peptide-recognition domain residues from protein sequence., *Bioinformatics*, bty352-bty352.

Iqbal, S., Mishra, A. and Hoque, T. (2015) Improved Prediction of Accessible Surface Area Results in Efficient Energy Function Application, *Journal of Theoretical Biology*, **380**, 380-391.

Islam, N., *et al.* (2016) A Balanced Secondary Structure Predictor, *Journal of Theoretical Biology*, **389**, 60-71.

Meiler, J., *et al.* (2001) Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks, *Molecular modeling annual*, **7**, 360-369.

Miao, Z. and Westhof, E. (2015) A Large-Scale Assessment of Nucleic Acids Binding Site Prediction Programs, *PLoS Comput Biol*, **11**, e1004639.

Mishra, A., Pokhrel, P. and Hoque, M.T. (2018) StackDPPred: a stacking based prediction of DNA-binding protein from sequence, *Bioinformatics*, **bty653**.

Mishra, A., Pokhrel, P. and Hoque, M.T. (2018) StackDPPred: a stacking based prediction of DNA-binding protein from sequence., *Bioinformatics*, bty653.

Nagi, S. and Bhattacharyya, D.K. (2013) Classification of microarray cancer data using ensemble approach, *Network Modeling Analysis in Health Informatics and Bioinformatics*, **2**, 159-173.

Pedregosa, F., *et al.* (2012) Scikit-learn: Machine learning in python., *Journal of Machine Learning Research*, **12**.

Pollastri, G., *et al.* (2002) Prediction of coordination number and relative solvent accessibility in proteins, *Proteins*, **47**, 147-153

Pollastri, G., *et al.* (2002) Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles, *Proteins*, **47**, 228-235.

Pruitt, K.D., *et al.* (2014) RefSeq: an update on mammalian reference sequences, *Nucleic Acids Research*, **42**, D756–D763.

Ren, H. and Shen, Y. (2015) RNA-binding residues prediction using structural features, *BMC Bioinformatics*, **16**.

Sharma, R., *et al.* (2018) OPAL: prediction of MoRF regions in intrinsically disordered protein sequences, *Bioinformatics*, **32**, 1850-1858.

Si, J., Zhao, R. and Wu, R. (2015) An Overview of the Prediction of Protein DNA-Binding Sites, *Int J Mol Sci*, **16**, 5194-5215.

Su, H., *et al.* (2019) Improving the prediction of protein–nucleic acids binding residues via multiple sequence profiles and the consensus of complementary methods, *Bioinformatics*, **35**, 930-936.

Szilágyi, A. and Skolnick, J. (2006) Efficient prediction of nucleic acid binding function from low-resolution protein structures., *Journal of Molecular Biology*, **358**, 922-933.

Terribilini, M., *et al.* (2007) RNABindR: a server for analyzing and predicting RNA-binding sites in proteins, *Nucleic Acids Research*, **35**, W578-W584.

Verma, R., Varshney, G.C. and Raghava, G.P.S. (2010) Prediction of mitochondrial proteins of malaria parasite using split amino acid composition and PSSM profile., *Amino Acids*, **39**, 101-110.

Walia, R.R., *et al.* (2016) Sequence-Based Prediction of RNA-Binding Residues in Proteins, *Methods in Molecular Biology*, **1484**, 205—235.

Wang, L. and Brown, S.J. (2006) BindN: a web-based tool for efficient prediction of DNA and RNA binding sites in amino acid sequences *Nucleic Acids Research*, **34**, W243-W248.

Wolpert, D.H. (1992) Stacked generalization, *Neural Networks*, **5**, 241-259.

Wong, K.-C. (2017) MotifHyades: expectation maximization for de novo DNA motif pair discovery on paired sequences, *Bioinformatics*, **33**, 3028-3035.

Yan, J., Friedrich, S. and Kurgan, L. (2015) A comprehensive comparative review of sequence-based predictors of DNA- and RNA-binding residues., *Briefings in Bioinformatics*, **17**, 88–105.

Yan, J. and Kurgan, L. (2017) DRNApred, fast sequence-based method that accurately predicts and discriminates DNA- and RNA-binding residues, *Nucleic Acids Research*, **45**, e84.

Yang, Y., *et al.* (2016) SPIDER2: A Package to Predict Secondary Structure, Accessible Surface Area, and Main-Chain Torsional Angles by Deep Neural Networks., *Methods in Molecular Biology*, **1484**.

Zhang, T., Faraggi, E. and Zhou, Y. (2010) Fluctuations of backbone torsion angles obtained from NMR-determined structures and their prediction, *PMC*, **78**, 3353-3362.

Zhang, T., *et al.* (2010) Analysis and Prediction of RNA-Binding Residues Using Sequence, Evolutionary Conservation, and Predicted Secondary Structure and Solvent Accessibility, *Current Protein and Peptide Science*, **11**, 609-628.

Zhang, X. and Liu, S. (2017) RBPPred: predicting RNA-binding proteins from sequence using SVM, *Bioinformatics*, **33**, 854-862.

Zhao, H., Yang, Y. and Zhou, Y. (2011) Structure-based prediction of RNA-binding domains and RNA-binding sites and application to structural genomics targets., *Nucleic Acids Research*, **39**, 3017-3025.

Altman, N.S. (1992) An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression, *The American Statistician*, **46**, 175-185.

Altschul, S.F., *et al.* (1990) Basic Local Alignment Search Tool., *J. Mol. Biol.*, **215**, 403-410.

Babu, M.M., *et al.* (2011) Intrinsically disordered proteins: regulation and disease., *Current Opinion in Structural Biology*, **21**, 432-440.

Bah, A. and Forman-Kay, J.D. (2016) Modulation of intrinsically disordered protein function by post-translational modifications., *Journal of Biological Chemistry*, **291**, 6696-6705.

Baltz, A.G., *et al.* (2012) The mRNA-Bound Proteome and Its Global Occupancy Profile on Protein-Coding Transcripts, *Molecular Cell*, **46**, 674-690.

Beckmann, B.M., *et al.* (2015) The RNA-binding proteomes from yeast to man harbour conserved enigmRBPs, *Nature Communications*, **6**.

Bergstra, J. and Bengio, Y. (2012) Random Search for Hyper-Parameter Optimization, *Journal of Machine Learning Research*, **13**.

Biswas, A.K., Noman, N. and Sikder, A.R. (2010) Machine learning approach to predict protein phosphorylation sites by incorporating evolutionary information., *BMC Bioinformatics*, **11**.

Breiman, L. (1996) Bagging predictors, *Machine Learning*, **24**, 123-140.

Calabretta, S. and Richard, S. (2015) Emerging roles of disordered sequences in RNA-binding proteins., *Trends in Biological Sciences*, **40**, 662-672.

Castello, A., *et al.* (2012) Insights into RNA Biology from an Atlas of Mammalian mRNA-Binding Proteins, *Cell*, **149**, 1393-1406.

Castello, A., *et al.* (2012) Insights into RNA biology from an atlas of mammalian mRNA-binding proteins., *Cell*, **149**, 1393-1406.

Dosztányi, Z., *et al.* (2005) The pairwise energy content estimated from amino acid composition discriminates between folded and intrinsically unstructured proteins, *Journal of Molecular Biology*, **347**, 827-839.

Dubchak, I., *et al.* (1995) Prediction of protein folding class using global description of amino acid sequence, *Proc. Natl. Acad. Sci. USA*, **92**, 8700-8704.

Dubchak, I., *et al.* (1995) Prediction of protein folding class using global description of amino acid sequence., *Proceedings of the National Acadecmy of Sciences of the United States of America*, **92**, 8700-8704.

Dubchak, I., *et al.* (1999) Recognition of a protein fold in the context of the SCOP classification., *Proteins*, **35**, 401-407.

Džeroski, S. and Ženko, B. (2004) Is Combining Classifiers with Stacking Better than Selecting the Best One?, *Machine Learning*, **54**, 255-273.

Glisovic, T., *et al.* (2008) RNA-binding proteins and post-transcriptional gene regulation, *FEBS Letters*, **582**.

Greenberg, J.R. (1979) Ultraviolet light-induced crosslinking of mRNA to proteins, *Nucleic Acids Research*, **6**, 715-732.

Han, L.Y., *et al.* (2004) Prediction of RNA-binding proteins from primary sequence by a support vector machine approach, *RNA*, **10**.

Hoque, M.T., *et al.* (2016) sDFIRE: Sequence-specific statistical energy function for protein structure prediction by decoy selections., *Journal of Computational Chemistry*, **37**, 1119-1124.

Järvelin, A.I., *et al.* (2016) The new (dis)order in RNA regulation., *Cell Communications and Signaling*, **14**.

Kumar, M., Gromiha, M.M. and Raghava, G.P. (2007) Identification of DNA-binding proteins using support vector machines and evolutionary profiles, *BMC Bioinformatics*, **8**, 1471-2105.

Kumar, M., Gromiha, M.M. and Raghava, G.P.S. (2008) Prediction of RNA binding sites in a protein using SVM and PSSM profile, *Proteins*, **71**.

Kumar, M., Gromiha, M.M. and Raghava, G.P.S. (2011) SVM based prediction of RNA-binding proteins using binding residues and evolutionary information., *Journal of Molecular Recognition*, **24**, 303-313.

Kumar, M., Gromiha, M.M. and Raghava, G.P.S. (2011) SVM based prediction of RNA-binding proteins using binding residues and evolutionary information, *Journal of Molecular Recognition*, **24**, 303-313.

Kwon, S.C., *et al.* (2013) The RNA-binding protein repertoire of embryonic stem cells., *Nature Structural & Molecular Biology*, **20**, 1122-1130.

Lina, Y.-H., *et al.* (2017) The intrinsically disordered N-terminal domain of galectin-3 dynamically mediates multisite self-association of the protein through fuzzy interactions., *Journal of Biological Chemistry*, **292**, 17845-17856.

Lindberg, U. and Sundquist, B. (1974) Isolation of messenger ribonucleoproteins from mammalian cells, *Journal of Molecular Biology*, **86**, 451-468.

Liu, S. RBPPred: Data sets updated.

Ma, X., Guo, J. and Sun, X. (2015) Sequence-based prediction of RNA-binding proteins using random forest with minimum redundancy maximum relevance feature selection., *BioMed Research International*, **425810**.

Ma, X., *et al.* (2015) PRBP: prediction of RNA-binding proteins using a random forest algorithm combined with an RNA-binding residue predictor., *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **12**, 1385-1393.

Magnan, C.N. and Baldi, P. (2014) SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity., *Bioinformatics*, **30**, 2592-2597.

Mishra, A. and Hoque, M.T. (2017) Three-Dimensional Ideal Gas Reference Sstate Based Energy Function, *Current Bioinformatics*, **12**, 171-180.

Mishra, A., Iqbal, S. and Hoque, M.T. (2016) Discriminate protein decoys from native by using a scoring function based on ubiquitous Phi and Psi angles computed for all atom., *Journal of theoretical biology*, **398**, 112-121.

Mitchell, S.F., *et al.* (2013) Global analysis of Yeast mRNPs., *Nature Structural & Molecular Biology*, **20**, 127-133.

Mohan, A., *et al.* (2006) Analysis of Molecular Recognition Features (MoRFs), *Journal of Molecular Biology*, **362**, 1043-1059.

Mohan, A., *et al.* (2006) Analysis of Molecular Recognition Features (MoRFs). *Journal of Molecular Biology*, **362**, 1043-1059.

Paz, I., *et al.* (2016) BindUP: a web server for non-homology-based prediction of DNA and RNA binding proteins., *Nucleic Acids Research*, **44**, W568-W574.

Sharma, R., *et al.* (2018) MoRFPred-plus: computational identification of MoRFs in protein sequences using physicochemical properties and HMM profiles., *Journal of Theoretical Biology*, **437**, 9-16.

Sharma, R., *et al.* (2018) OPAL+: length-specific MoRF prediction in intrinsically disordered protein sequences., *Proteomics*, **1800058**.

Shazman, S. and Mandel-Gutfreund, Y. (2008) Classifying RNA-binding proteins based on electrostatic properties., *PLoS Computational Biology*, **4**.

Shen, J., *et al.* (2007) Predicting protein–protein interactions based only on sequences information., *Proceedings of the National Academcy of Sciences of the United States of America*, **104**, 4337-4341.

Si, J., *et al.* (2015) Computational Prediction of RNA-Binding Proteins and Binding Sites, *International Journal of Molecular Sciences*, **16**, 26303-26317.

Vacic, V., *et al.* (2007) Characterization of molecular recognition features, MoRFs, and their binding partners, *J Proteome Res.*, **6**, 2351-2366.

Wagenmakers, A.J.M., Reinders, R.J. and Venrooij, W.J.V. (1980) Cross-linking of mRNA to Proteins by Irradiation of Intact Cells with Ultraviolet Light, *European Journal of Biochemistry*, **112**.

Wang, Y., *et al.* (2013) De novo prediction of RNA–protein interactions from sequence information., *Molecular BioSystems*, **9**, 133-142.

Wong, K.-C. (2017) MotifHyades: expectation maximization for de novo DNA motif pair discovery on paired sequences, *Bioinformatics*, **33**, 3028-3035.

Wu, C.H., *et al.* (2006) The Universal Protein Resource (UniProt): an expanding universe of protein information, *Nucleic Acids Research*.

Wurth, L. (2012) Versatility of RNA-Binding Proteins in Cancer, *International Journal of Genomics*, **2012**, 178525.

Xu, R., *et al.* (2015) Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation, *BMC Systems Biology*, **9**.

Yang, Y., *et al.* (2012) A new size-independent score for pairwise protein structure alignment and its application to structure classification and nucleic-acid binding prediction., *Proteins*, **80**, 2080-2088.

Zhang, L., Zhao, X. and Kong, L. (2014) Predict protein structural class for low-similarity sequences by evolutionary difference information into the general form of Chou′s pseudo amino acid composition., *Journal of Theoretical Biology*, **355**, 105-110.

Zhao, H., Yang, Y. and Zhou, Y. (2011) Highly accurate and high-resolution function prediction of RNA binding proteins by fold recognition and binding affinity prediction., *RNA Biology*, **8**, 988-996.

Zhou, H. and Skolnick, J. (2011) GOAP: A Generalized Orientation-Dependent, All-Atom Statistical Potential for Protein Structure Prediction., *Biophys. J* . **101**, 2043-2052.