sRNA-Protein Interaction Prediction in Bacteria

by

© Atiyeh Tahavorgar

A Thesis Report submitted to the

School of Graduate Studies

in partial fulfillment of the

requirements for the degree of

Master of Science

Supervisor: Dr. Lourdes Peña-Castillo

Department of Computer Science

Memorial University of Newfoundland

August 2023

St. John's Newfoundland

Abstract

In bacteria, many biological processes such as stress response, metabolism, and post-transcriptional gene expression regulation are mediated by interactions of proteins with small RNAs (sRNAs). sRNAs are non-coding RNAs (ncRNAs) between 50 to 500 nucleotides long [1]. There are several experimental or wet-lab approaches to determine sRNA-protein interactions; however, wet-lab methods are expensive, time-consuming, and labor-intensive. Computational approaches, on the other hand, once developed, can predict sRNA-protein interactions quickly and affordably.

Current RNA-protein interaction prediction methods have been generated using data from a variety of RNAs (mRNAs, lnRNAs, ncRNAs, etc) and organisms (mammals, bacteria, plants). We hypothesized that a model generated specifically with experimentally validated interacting bacterial sRNA-protein pairs would have a better performance in predicting bacterial sRNA-protein interactions than current methods. To do that, we collected from the literature roughly 1.5k experimentally determined interacting sRNA-protein pairs and used these data to train various machine-learning approaches. Using cross-validation, we selected the most accurate model. Our model achieves an average accuracy of 0.885 ±0.03 on four commonly used RNA-protein interaction data sets which are comparable to other methods. However, we were unable to confirm our initial hypothesis as ProNA's performance was not better than that of other methods in predicting bacterial sRNA-protein interactions.

Contents

List of Figures

List of Tables

List of Abbreviations

ANOVA Analysis of variance

AUC Area under the ROC curve

AUROC Area under the ROC curve

BED Browser extensible data

BLSTM Bidirectional long short-term memory

cDNA Complementary DNA

CNN Convolutional neural network

CSV Comma-separated values

CTF Conjoint triad function

CV Cross-validation

DNA Deoxyribonucleic acid

DSAN Deep stacking autoencoder network

DT Decision tree

ENB Extended naïve bayes

ExtraTree Extremely randomized tree

GCN Graph convolutional network

GFF General feature format

GNN Graph neural network

GUI Graphical user interface

LM Legendre moments

lnRNA Long noncoding RNA

LR Logistic regression

MCC Matthew's correlation coefficient

MI Mutual information

mRNA Messenger ribonucleic acid

NB Naïve bayes

ncRNA Non-coding RNA

ncRPI Non-coding RNA-protein interaction

NDB Nucleic acid database

NLP Natural language processing

PB Protein block

PDB Protein data bank

PPV Positive predictive value

PR Precision recall

PRIDB Protein-RNA interface database

PSSM Position specific scoring matrix

PWM Position weight matrix

PZM Pseudo-zernike moment

RBP RNA binding protein

RF Random forest

RNA Ribonucleic acid

ROC Receiver operating characteristic

RPI RNA-protein interaction

RPKM Reads per kilobase of a transcript, per million mapped reads

RSS RNA secondary structure

SAE Stacked auto-encoder

sRNA Small RNA

sRPI sRNA-protein interaction

SVD Singular value decomposition

SVM Support vector machine

TNR True negative rate

TPR True positive rate

TSV Tab-separated values

tSVD Truncated singular value decomposition

UniProt Universal protein resource

VCF Variant call format

XGBoost Extreme gradient boosting

Chapter 1

Introduction

sRNA-protein interaction (sRPI) plays a crucial role in post-transcriptional regulation in bacteria [6]. Understanding these interactions is essential, among other things, to realize how bacteria respond to environmental stimuli and bacterial pathogenesis. Recent high-throughput sequencing techniques have identified the sRNA partners of several proteins, such as Hfq [14], CsrA [15], ProQ [16], and FinO [17].

There are two approaches for predicting RNA-protein interactions: interface prediction and partner prediction. The interface prediction approach detects the amino acid residues in a protein that are expected to bind with an RNA [6]. Partner prediction is the recognition of specific RNA interaction partner(s) for a known RNA binding protein [6]. This project focused on sRNA-protein (partner) interaction prediction in bacteria using only sequence-derived features as protein and sRNA sequences are widely available. Unlike other partner prediction studies that have used data from different organisms (e.g. animals, fungi, and bacteria) and different kinds of RNAs (e.g. ncRNA and mRNA), we only considered sRNA-protein interactions in bacteria. In this study, after collecting experimentally-determined sRNA-protein interactions from the literature, we calculated sequenced-derived features to represent sRNA and protein sequences in our data set. Next, we selected a subset of these features using feature selection methods. Then, we generated and assessed the performance of several machine-learning methods for predicting sRNA-protein interaction. Finally, we implemented our best model in ProNA, a protein-sRNA interaction predictor. ProNA achieved an accuracy of 0.885 ±0.03 on four data sets commonly used by previous studies. The remaining of the thesis is organized as follows:

- Chapter 2: Molecular Biology and Computational Background
- Chapter 3: Methodology
- Chapter 4: Results and Discussion
- Chapter 5: Conclusion

Chapter 2

Background and Related Works

2.1 Molecular Biology Basics of RNA-Protein Interactions

RNA-protein interactions are crucial for various cellular processes, including gene expression regulation, RNA processing, transport, and translation[18]. Understanding the basics of these interactions is essential for comprehending the molecular mechanisms underlying these processes. Here's an overview of the key concepts in RNA-protein interactions [18]:

1. RNA molecules:

   - Messenger RNA (mRNA): Carries the genetic information from DNA to the ribosome, where it serves as a template for protein synthesis.

   - Transfer RNA (tRNA): Brings amino acids to the ribosome during translation and helps in assembling polypeptide chains.

- Ribosomal RNA (rRNA): Major component of ribosomes, where protein synthesis occurs.
- sRNAs are small noncoding RNAs.

2. RNA-binding proteins (RBPs):

   - RBPs are a diverse group of proteins that recognize and interact with RNA molecules. They contain specific RNA-binding domains that allow them to bind to RNA sequences or structures (Fig. 2.1).
   - Some RBPs are general, associating with various RNA molecules, while others are highly specific, recognizing particular RNA targets.

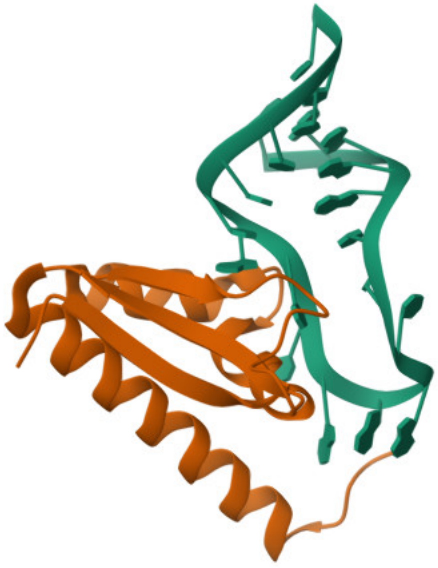


Figure 2.1: RNA-binding domain (in red) interacting with a 20-nucleotide RNA in a hairpin structure (in green). This figure is sourced from the RCSB Protein Data Bank (RCSB PDB) website (RCSB.org) for the entry 1EC6 [2].

3. RNA secondary structures:
    - RNA molecules can fold into intricate secondary structures due to complementary base pairing.
    - Common secondary structures include hairpins, loops, stems, and bulges. These structures play a significant role in RBP recognition and binding.
4. RBP-RNA recognition:
    - RBPs recognize specific RNA sequences or structural motifs through their RNA-binding domains.

- Recognition can involve hydrogen bonding, van der Waals interactions, electrostatic interactions, and hydrophobic interactions between amino acids of the RBP and nucleotides of the RNA.

5. Functions of RNA-protein interactions:

   - Stabilization and protection: RBPs can protect RNA molecules from degradation by forming complexes with them.
   - Transport: RBPs facilitate the transport of specific RNA molecules to their subcellular destinations.
   - Translation: Initiation factors and ribosomal proteins interact with mRNA and tRNA to regulate translation. Post-transcriptional regulation: miRNAs and RBPs influence mRNA stability and translation efficiency.

2.2 Related Works

In this section, we provide an overview of previous RNA-protein interaction prediction methods. Most of the researchers have used machine-learning methods for their predictors, such as RPI-Pred [8], RPISeq [6], and RPI-SE [12]

In recent years, there have been several publications on the computational prediction of RNA-protein interactions. Muppirala et al. [6] introduced a method named RPISeq, which used random forest (RF) and support vector machine (SVM) classifiers based on primary sequence information. Afterward, Xiaowei et al. [7] used Naive-

Bayes (NB) and Extended Naive-Bayes (ENB) classifiers with sequence information for RPIs prediction. In 2015, Suresh et al. [8] presented RPI-Pred, a computational approach based on a support vector machine (SVM) classifier to predict RPIs by using both sequences and high-order structure information. Hai-Cheng et al. [9] proposed a model (RPI-SAN) with deep learning stacked auto-encoder network to mine the hidden high-level features from RNA and protein sequences and feed them into a random forest (RF) model for RNA-protein interaction prediction. RPiRLS was created by Shen et al. [10] to predict ncRPI with sequence information. In 2019 Cheng et al. [3] made DM-RPIs for predicting ncRNA-protein interactions with sequence-derived information. RPITER was built by Peng et al. [11] with deep learning to predict RPI considering sequence and structure features. Zhu-Hong et al. [12] created a model (RPI-SE) based on a support vector machine (SVM) classifier to predict ncRNA-protein interaction using just sequence information from ncRNA as well as protein sequences. Recently, Zhao et al. [4] created EDLMFC with deep learning to predict ncRNA-protein interactions using sequence-derived and structure-derived information. In 2022, Ren et al. [5] made SAWRPI to predict ncRNA-protein interactions by considering only sequence information. Arora et al. [13] made a deep learning model with sequence-based features to predict the whole RPI network.

The rest of this chapter will describe all of these methods in more detail. Moreover, different datasets were used for training and testing the methods which are defined in each approach.

2.2.1 RPISeq

RPISeq [6] is a method that uses a support vector machine and random forest classifiers. For training, RNA-protein interacting pairs were extracted from 943 protein-RNA complexes (containing a total of 9,689 proteins and 2,074 RNAs) in PRIDB [19]. These protein-RNA complexes form the RPI2241 and RPI369 data sets. PRIDB is a database of protein-RNA interactions calculated from protein-RNA complexes in the protein data bank (PDB) [20]. Researchers randomly paired the RNAs and proteins from the 943 protein-RNA complexes and removed interacting RNA-protein pairs to generate a negative data set that contains non-interacting RNA-protein pairs.

For evaluating the method on independent RPI data sets, the following data sets were used:

- 5,166 mRNA-protein interactions [21]
- 13,243 RPIs, which include all 5,166 interactions in the previous data set [21]
- NPInter database [22] for predicting ncRNA-protein interaction networks

In this approach, each RNA-protein pair is represented as a 599-feature vector. 343 $(7 \times 7 \times 7)$ features are used to encode the protein sequence with the conjoint triad function (CTF) method. The CTF representation essentially encodes each protein sequence using the normalized 3-gram (3-amino acid) frequency distribution extracted from a 7-letter reduced alphabet representation of the protein sequence. Next, 256

$(4 \times 4 \times 4 \times 4)$ features are used to encode the RNA sequence using normalized tetranucleotide frequencies extracted directly from the 4-letter ribonucleotide alphabet representation of the RNA sequence.

Table 2.1 shows the 10-fold cross-validation (CV) performance of random forests (RF) and support vector machines (SVMs) on the RPI2241 and RPI369 data sets.

Metric	**RPI2241-RF**	**RPI2241-SVM**	**RPI369-RF**	**RPI369-SVM**
Accuracy(%)	89.6	87.1	76.2	72.8
Precision	0.89	0.87	0.75	0.73
Recall	0.90	0.88	0.78	0.73
F-measure	0.90	0.87	0.77	0.73

Table 2.1: RPISeq performance summary as reported by [6].

2.2.2 De novo prediction of RNA–protein interactions from sequence information

This method [7] same as [6] uses only features from RNA and protein sequences without requiring any structure-derived information. Naive Bayes (NB) and Extended Naive Bayes (ENB) were implemented for RNA-protein interaction prediction:

- NB classifier is a fast and effective learning approach for predicting protein–RNA interactions, which follows the assumption of the independence between the features;
- ENB classifier takes the feature dependency into account and thus is able to

offer accurate prediction with correlated features.

Positive sample sets (RNAs and proteins that can interact with each other) consist of 367 interacting pairs of ncRNA and protein. Negative sample sets were constructed by randomly pairing the RNA and protein sequences after removing the pairs that existed in the positive sample sets. All the features were combined to form the feature vector, which was approximately a

$$4 \times 4 \times 4 \times 4^k$$

RNA-protein interactions. K denotes the nucleotide acids CTF for RNA sequences. In Tables 2.2 and 2.3 the 10-fold CV performance of the NB and the ENB classifiers, with 1000 features, are shown:

Metric	**RPI2241**	**RPI369**	**NPInter**
Accuracy	0.73	0.74	0.74
Sensitivity	0.41	0.36	0.35
Specificity	0.89	0.94	0.93
Precision	0.65	0.75	0.73
MCC	0.35	0.39	0.37

Table 2.2: Prediction results of the NB classifier as reported by [7].

Metric	**RPI2241**	**RPI369**	**NPInter**
Accuracy	0.74	0.75	0.77
Sensitivity	0.38	0.34	0.47
Specificity	0.91	0.95	0.92
Precision	0.69	0.77	0.76
MCC	0.36	0.39	0.46

Table 2.3: Prediction results of the ENB classifier as reported by [7].

2.2.3 RPI-Pred

RPI-Pred [8] method uses SVM (LibSVM package [23] and polynomial kernel) classifier. For developing this method, a non-redundant training data set of RPI complexes was collected by getting the Nucleic Acid Database (NDB) [24] and the protein-RNA interface database (PRIDB) [13]. NDB [24] provides data for RNA-protein complexes, whereas the PRIDB [19] provides atomic interfaces for RNA-protein interacting pairs. As of 1 February 2014, 1560 RPI complexes from NDB were used. Also, atomic and chain interfaces for 1336 complexes from PRIDB, which consist of both positive and negative protein-RNA pairs were utilized. RPI369, RPI2241, and NPInter10412 (a subset of NPInter database v2.0 containing 10,412 ncRPI pairs of six model organisms) [22] data sets were used for testing.

In addition to sequences, experimentally-determined structures were also utilized:

- A protein 3D structure represented by 16-letter 1D structural fragments, called PBs (Protein Blocks). The PDB-2-PB database [25] provides the PB information based on the experimentally solved protein structures available in PDB. They used the PDB-2-PB database to retrieve the 16-letter PB structure features for each protein in the training data set.

- 3DNA suite [26] was employed to extract the RSS (RNA Secondary Structure) from the corresponding 3D structures for each RNA in the training data set.

After that, 112 protein features were obtained by combining 7 amino acid groups times 16 protein blocks. Also, 20 RNA features were gathered by combining 4 nucleotides (A, C, G, T) times five RNA secondary structures (stem, hairpin, loop, bulge, and internal loop). In sum, 132 features were utilized to encode RNA-protein interacting pairs (20 features for RNAs and 112 features for proteins).

To predict protein blocks (PBs) and RNA secondary structures (RSS) in test data sets, the following libraries were used:

- PB-kPRED method [26] was used to predict the protein block structures for proteins

- RNAfold from the Vienna package [27] was applied to predict the RNA secondary structure for RNAs

Table 2.4 illustrates the performance of RPI-Pred using a 10-fold CV:

Metric	**RPI2241**	**RPI369**	**NPInter10412**
Precision	0.88	0.89	0.85
Recall	0.78	0.89	0.90
F-score	0.83	0.89	0.87
Accuracy	0.84	0.92	86.9

Table 2.4: RPI-pred performance summary as reported by [8].

2.2.4 RPI-SAN

RPI-SAN [9] (a sequence-based approach) uses a deep learning stacked auto-encoder network to mine the hidden features of RNA and protein sequences. These features were then passed to a random forest (RF) classifier. RPI2241, RPI488, RPI1807, and NPInter v2.0 data sets were employed. First, RNA sequences were converted into a k-mers sparse matrix [27]. Then the singular value decomposition (SVD) was used to extract the feature vector for each sequence [28]. Regarding protein sequences, a pseudo-Zernike moment (PZM) descriptor [29] was used for extracting the evolutionary information from the position-specific scoring matrix (PSSM). Finally, these features were employed to the stacked auto-encoder and random forest for learning features and predicting RPIs, respectively.

Mentioned data sets were used with a 5-fold CV procedure. The result of the current method on RPI2241, RPI488, and RPI1807 data sets is shown in Table 2.5, and researchers found that the accuracy of RPI-SAN is 98.67% on the independent data set NPInter v2.0 [22].

Metric	**RPI2241**	**RPI488**	**RPI1807**
Accuracy(%)	90.77	89.7	96.1
Sensitivity(%)	86.17	94.3	93.6
Specificity(%)	97.37	83.7	99.9
Precision(%)	84.05	95.2	91.4
MCC(%)	82.27	79.3	92.4
AUC	0.962	0.920	0.999

Table 2.5: Prediction results of the RPI-SAN classifier as reported by [9].

2.2.5 RPiRLS

This method [10] is a machine-learning model that combines a sequence-based derived kernel (extracts the contextual information around an amino acid or a nucleic acid as well as the repetitive conserved motif information) with regularized least squares [30]. There are 2 versions of RPiRLS, RPiRLS, and RPiRLS-7G. In RPiRLS each protein sequence comprises up to 20 diverse amino acids but in RPiRLS-7G each protein sequence is represented by using 7-letter reduced alphabets based on their physiochemical properties.

RPiRLS and RPiRLS-7G classifiers were trained on the RPI2662 data set, and tested on the RPI2241 and RPI369 data sets. Tables 2.6 and 2.7 show the performance of classifiers on the RPI369 and the RPI2241 data sets, respectively.

Metric	**RPiRLS**	**RPiRLS-7G**
Accuracy	0.85	0.79
AUC	0.92	0.90
Specificity	0.84	0.72
Sensitivity	0.86	0.87

Table 2.6: RPI369 10-fold CV results as reported by [10].

Metric	**RPiRLS**	**RPiRLS-7G**
Accuracy	0.80	0.67
AUC	0.80	0.74
Specificity	0.82	0.58
Sensitivity	0.79	0.76

Table 2.7: RPI2241 10-fold CV results as reported by [10].

2.2.6 DM-RPIs

Deep mining ncRNA-protein interactions (ncRPIs) [3] is a classifier that was trained to predict ncRPIs from sequence information. Three methods were used for training:

1. Random Forest (RF)

2. Support Vector Machine (SVM)

3. Convolution Neural Network (CNN)

These classifiers were then merged with the stacked ensemble method. DSANs were trained to preprocess raw data. 20 amino acids were divided into 7 groups according to their dipole moments and the volume of their side chain. Each protein sequence was represented by conjoint triad features (CTF). CTF shows a normalized frequency of 3-mer in the 7-letter representation of the protein sequence, resulting in 343 (7×7×7) dimensional features. Each RNA chain was represented by the normalized frequency of the 4-mer sequence fragment, which made 256 (4×4×4×4) dimensional features. A vector of 599 (343+256) dimensions represented each interaction. Then, DSANs (Deep Stacking Auto-encoders Networks) model was used to reduce the dimension of the features to 128.

Five data sets were utilized for training using a 5-fold CV: RPI369, RPI488, RPI1807, RPI2241 and RPI13254. Figures 2.2 and 2.3 show the performance of the 4 mentioned classifiers on RPI369 and RPI2241, respectively.

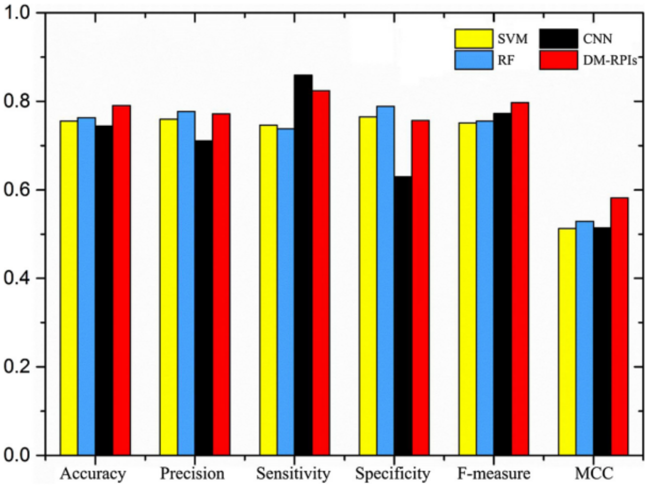


Figure 2.2: Comparison of three base classifiers and DM-RPIs on RPI369. Reprinted from [3] page 107088, Copyright (2019), with permission from Elsevier.

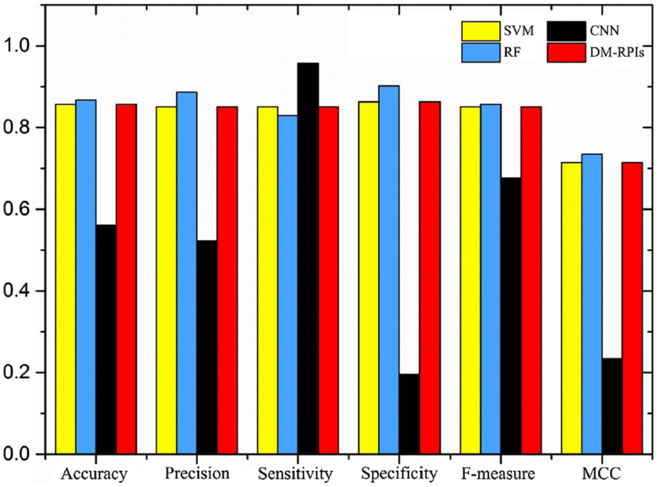


Figure 2.3: Comparison of three base classifiers and DM-RPIs on RPI2241. Reprinted from [3] page 107088, Copyright (2019), with permission from Elsevier.

2.2.7 RPITER

RPITER [11] (hierarchical deep learning-based framework) uses sequence as well as structure information taken from RNA and protein sequences. RPITER utilized two basic neural network architectures of convolution neural network (CNN) and stacked auto-encoder (SAE). The same CTF procedure of [3] (previous method) was applied, and they got a 599 (343+256) dimensions feature vector for all interaction pairs. Besides CTF, 3 deep learning sequence coding methods were employed: one hot, word2vec, and doc2vec. After comparing 4 sequence coding approaches, CTF had better results in comparison with 3 other techniques. The structure information of the protein and RNA sequences were predicted with SOPMA [31] and viennaRNA [27], respectively. Moreover, cd-hit [32] was applied to cluster RNA and protein sequences with an identity threshold of 0.4.

RPI369, RPI488, RPI1807, RPI2241, and NPInter data sets were utilized in this study with a 5-fold CV procedure. Table 2.8 shows the performance of RPITER on different data sets.

Metric	**RPI369**	**RPI488**	**RPI1807**	**RPI2241**	**NPInter**
Accuracy	0.72	0.89	0.96	0.89	0.95
Sensitivity	0.79	0.83	0.98	0.91	0.97
Specificity	0.0.65	0.94	0.94	0.86	0.93
Precision	0.70	0.94	0.95	0.87	0.93
MCC	0.46	0.79	0.93	0.78	0.91
AUC	0.82	0.91	0.99	0.95	0.98

Table 2.8: RPITER classifier results as reported by [11].

2.2.8 RPI-SE

RPI-SE [12] is a stacking ensemble computational framework that uses three base classifiers: Gradient Boosting Decision Tree, SVM, and Extremely Randomized Trees (ExtraTree) to predict ncRNA-protein interactions via sequence information. The output of these three base classifiers was then combined with Logistic Regression (LR).

ncRNA and protein sequences were represented as follows:

- RNA sequences: K-mer sparse matrix was implemented. It scanned each RNA sequence (A, C, G, U) with a k (k=4 for RNA) nucleotide window, moving one nucleotide at a time. After that, singular value decomposition (SVD) was implemented to reduce the matrix into a 256-feature vector.

- Protein sequences: position weight matrix (PWM) was employed. It has one row for each symbol of the alphabet and 20 rows for amino acids in protein sequences. Next, Legendre Moments (LMs) [33] feature vectors were extracted from the PWM of protein sequences (676 feature vectors). Then, truncated singular value decomposition (tSVD) was performed to reduce the influence of noise with 500 feature vectors.

Finally, each pair of ncRNA-protein was represented with 756 features. Three data sets, RPI369, RPI488, and RPI1807, were utilized to evaluate the performance of RPI-SE with a 5-fold CV procedure.

Table 2.9 depicts the result of this classifier on 3 data sets.

Metric	**RPI369**	**RPI488**	**RPI1807**
Accuracy (%)	88.44	89.30	96.86
TPR (%)	83.69	94.49	96.71
TNR (%)	95.87	83.48	97.69
PPV (%)	80.85	95.15	95.83
MCC (%)	77.73	79.31	93.65
AUC	0.92	0.90	0.99

Table 2.9: RPI-SE classifier results on 3 data sets as reported by [12].

2.2.9 EDLMFC

EDLMFC [4] is a deep learning-based method, to predict ncRNA–protein interactions using primary sequence features, secondary structure sequence features, and tertiary structure features. CNN (Convolutional neural network) and BLSTM (Bi-directional long short-term memory network) were used. The Conjoint k-mer (3-mer frequency feature for proteins and 4-mer frequency feature for ncRNAs) method was used to extract features. RPI1807, NPInter, and RPI488 data sets were used. To analyze the contributions of the three kinds of features, seven different feature combinations were created:

1. sequence
2. secondary structure
3. tertiary structure
4. sequence together with secondary structure

5. sequence together with tertiary structure

6. secondary structure together with tertiary structure

7. all features together

Figure 2.4 represents the contribution of the 7 mentioned combinations on the RPI1807 data set.

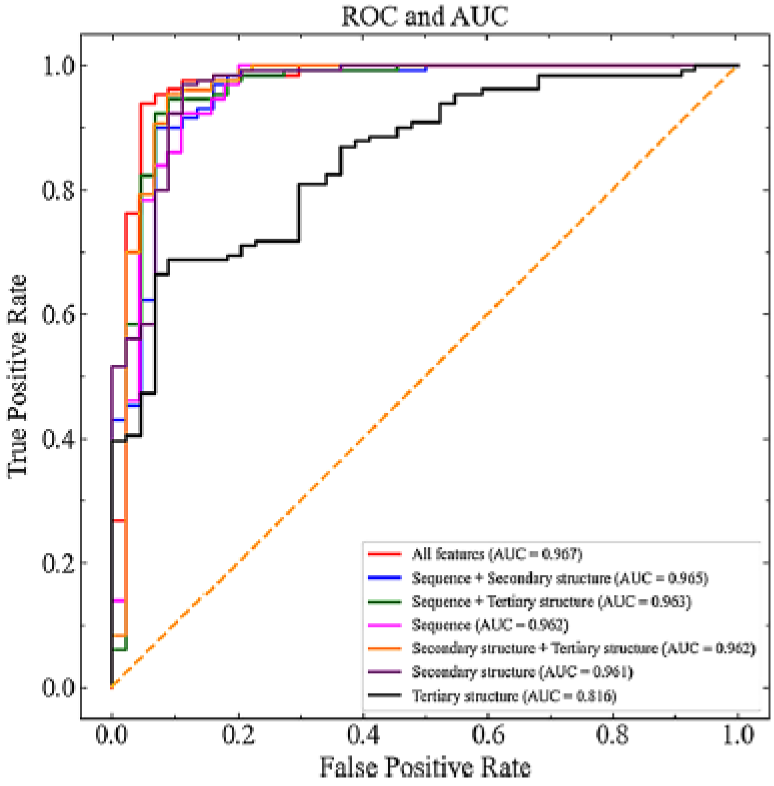


Figure 2.4: Contribution of 7 feature combinations on RPI1807. Figure taken from [4] (CC BY 4.0).

Table 2.10 shows the performance of EDLMFC on three data sets with a 5-fold CV procedure.

Metric	**RPI1807**	**NPInter**	**RPI488**
Accuracy(%)	93.8	89.7	86.1
F1-score(%)	95.9	89.9	82.9
MCC(%)	83.3	79.5	74.2
AUC(%)	96.7	95.9	89.9
TPR(%)	96.9	91.7	74.5
TNR(%)	84.5	87.7	96.7
PPV(%)	94.9	88.2	96.1

Table 2.10: EDLMFC 5-fold CV results as reported by [4].

EDLMFC classifier was further evaluated on an independent data set to check whether ncRNAs interact with proteins or not. RPI1807 and NPInter data sets were used to train and test the model, respectively. Table 2.11 compares the actual number of ncRNA–protein pairs in NPInter and EDLMFC's accuracy.

Organism	**NPInter pairs**	**EDLMFC accuracy**
Homo sapiens	740	631 (85%)
Mus musculus	229	217 (95%)
Saccharomyces cerevisiae	693	632 (91%)
Caenorhabditis elegans	33	31 (94%)
Drosophila melanogaster	46	41 (89%)
Escherichia coli	202	188 (93%)
Total	1943	1742 (90%)

Table 2.11: EDLMFC performance on different organisms as reported by [4].

2.2.10 SAWRPI

SAWRPI [5] was proposed to make a prediction of ncRNA-protein through sequence information. This method integrates four base classifiers XGBoost, SVM, ExtraTree and Random Forest for classification and prediction. The stacking ensemble was used to integrate 4 base classifiers. They got information on amino acids through a 3-mers sparse matrix and then generated a feature vector through SVD. For ncRNA representation, natural language processing (NLP) was used to retrieve a representation of ncRNA nucleic acid symbols, then get comprehensive information through a local fusion strategy. To make the classification easier, Hilbert Transformation was exploited to feature extraction which transformed raw feature data into a new feature space. RPI369, RPI1807, and RPI488 data sets were used with a 5-fold CV. Table 2.12 demonstrated the performance of SAWRPI on these 3 data sets.

Metric	**RPI369**	**RPI488**	**RPI1807**
Accuracy	0.71	0.89	0.96
Precision	0.69	0.93	0.96
Sensitivity	0.75	0.84	0.98
F1-score	0.72	0.88	0.97
MCC	0.42	0.79	0.93

Table 2.12: SAWRPI performance on three data sets as reported by [5].

Moreover, Figure 2.5 shows the ROC curve of SAWRPI and 5 classifiers. In figure 2.5, different classifiers show different percentages for distinguishing between classes.

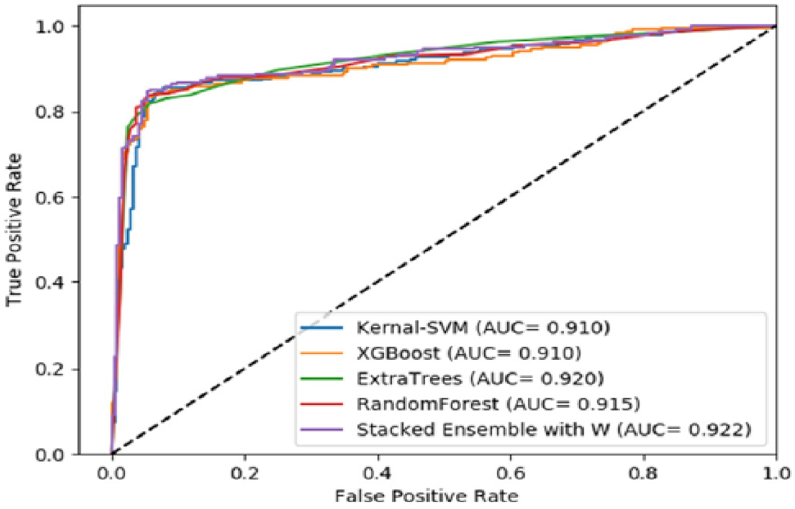


Figure 2.5: ROC curve of 6 classifiers on RPI1807. Figure taken from [5] (CC BY 4.0).

2.2.11 De novo prediction of RNA-protein interactions with graph neural networks

Graph Convolutional Network (GCN) [13] with 2 convolutional layers as a GNN model was proposed. CLIP-seq data was used to retrieve RBPs and a set of RNAs, that a protein can bind to. To construct the benchmark data sets, they used the eCLIP data set for two cell lines (HepG2 and K562):

1. HepG2: 15018 nodes (103 proteins and 14915 RNAs) and 145509 interactions (edges)

2. K562: 14665 nodes (120 proteins and 14545 RNAs) with 144527 interactions between proteins and RNAs.

For extracting sequence features, the following methods were used:

1. Proteins: conjoint triad descriptors extract the features based on their dipoles and volumes of the side chains. Each protein sequence was encoded from a 7-letter reduced alphabet representation.

2. RNAs: 6-mer frequency distribution was taken from each RNA sequence.

The above feature extraction methods were used to generate 7^3 (343) and 4^6 (4096) dimensional feature vectors for protein and RNAs, respectively. Tables 2.13 and 2.14 compare the AUROC (area under the receiver operating characteristic) of the HepG2 cell line and K562 cell line with RNAcommender (another method)[34] and RPIseq methods with 10-fold CV procedures, respectively.

Test set with different percent of edges	**RNAcommender**	**RPIseq**	**GCN**
10%	0.632 ±0.004	0.808 ±0.003	0.771 ±0.003
20%	0.628 ±0.004	0.799 ±0.002	0.762 ±0.003
30%	0.621 ±0.004	0.791 ±0.001	0.796 ±0.003
40%	0.618 ±0.004	0.782 ±0.002	0.742 ±0.004
50%	0.618 ±0.004	0.773 ±0.001	0.734 ±0.001

Table 2.13: GCN (HepG2) performance comparison with 2 other methods as reported by [13].

Test set with different percent of edges	**RNAcommender**	**RPIseq**	**GCN**
10%	0.855 ±0.003	0.868±0.002	0.926 ±0.002
20%	0.852 ±0.003	0.865 ±0.002	0.921 ±0.001
30%	0.846 ±0.003	0.857 ±0.001	0.911 ±0.002
40%	0.844 ±0.003	0.857 ±0.001	0.909 ±0.002
50%	0.841 ±0.005	0.854 ±0.001	0.904 ±0.001

Table 2.14: GCN (K562) performance comparison with 2 other methods as reported by [13].

2.2.12 RNA-protein prediction methods overview

Table 2.15 provides an overview of all of the related works.

Application	**Year**	**Approach**	**Data used**	**Prediction approach**	**Ref**
RPISeq	2011	SVM and RF	sequence information	Partner	[6]
De novo prediction of RNA–protein interactions from sequence information	2013	NB and ENB	sequence information	Partner	[12]
RPI-Pred	2015	SVM	sequence and structural information	Partner	[8]
RPI-SAN	2018	Stacked Autoencoder and RF	sequence information	Interface	[9]
RPiRLS	2018	Regularized Least Squares	sequence information	Partner	[10]
DM-RPIs	2019	SVM and RF and CNN	sequence information	Interface	[3]

Application	**Year**	**Approach**	**Data used**	**Prediction approach**	**Ref**
RPITER	2019	CNN and SAE	sequence and structural information	Partner	[3]
RPI-SE	2020	SVM and XGBoost and ExtraTree	sequence information	Partner	[12]
EDLMFC	2021	CNN and BLSTM	sequence and structural information	Interface	[4]
SAWRPI	2022	XGBoost and SVM and ExtraTree and RF	sequence information	Partner	[5]
De novo prediction of RNA-protein interactions with graph neural networks	2022	Graph Convolutional Network	sequence information	Partner	[13]

Table 2.15: Related works overview

2.3 Feature Extraction Methods

Most of the methods described in Section 2.1 used RNA and protein sequence features. Recently, several methods for extracting sequence-derived features have become available. Here we present 13 different Python-based programs to extract features from sequences.

2.3.1 propy

propy [35] (2013) extracts PseAAC (pseudo amino acid composition) descriptors from proteins and peptide sequences. It extracts 13 different features. This method is suitable for protein sequences.

2.3.2 PyDPI

PyDPI (drug-protein interaction with Python) [36] is a freely available Python package that can be applied only to protein sequences for extracting 14 features. PyDPI emphasizes on integration of chemoinformatics and bioinformatics into a molecular informatics platform for drug discovery.

2.3.3 SPiCE

SPICE [37] (2014) extracts sequence-based features from protein sequences. This package extracts 17 different features. It also provides easy access to visualization and classification methods for a set of protein sequences. It can run on Chrome, Firefox, Opera, and Safari browsers.

2.3.4 PseKNC-General

This package [38] (the general form of pseudo-k-tuple nucleotide composition) was developed in 2014. It extracts RNA and DNA features from their sequences. This package:

1. Can be run on Linux, Mac, and Windows systems
2. Provides a graphical user interface

2.3.5 ProFET

ProFET [39] (Protein Feature Engineering Toolkit) was built in 2015 and extracted 24 features. Unfortunately in this method, some features cannot be obtained directly from the sequence.

2.3.6 repDNA

repDNA [40] is a Python package that is used for extracting features from DNA and nucleotide sequences. This package was published in 2014 and is able to extract 15 features.

2.3.7 POSSUM

This tool [41] was published in 2017 to extract 21 features from protein sequences. POSSUM is an online web server that can generate features based on PSSM (Position-

Specific Scoring Matrix).

2.3.8 iFeature

This open-source Python package (2018) [42] can find 53 different types of feature descriptors from protein and peptide sequences. iFeature is also freely available via an online web server and a stand-alone program.

2.3.9 PyBioMed

This is a Python package that was published in 2018 [43]. It can extract 28 different features from protein and DNA sequences by providing various user-friendly and highly customized APIs, 14 features for each of them. This Python package is open-access and can run on Linux and Windows operating systems.

2.3.10 BioSeq-Analysis

BioSeq-Analysis [44] can be used with RNA, DNA, and protein sequences to produce different features via 56 feature extraction methods. 20, 14, and 22 methods for DNA, RNA, and proteins, can be calculated, respectively. This Python package can run as a stand-alone program and a web server. The program can be directly run on Windows, Linux, and UNIX.

2.3.11 PyFeat

PyFeat [45] is a feature extraction tool that can be used for RNA, DNA, and protein sequences. This package successfully extracts 13 features from DNA and RNA sequences, as well as 9 features from RNA, DNA, and protein sequences together. PyFeat Comes with a dimension reduction method to reduce the dimensionality of extracted features.

2.3.12 iLearnPlus

This open-source package [46] can extract multiple feature sets from RNA, DNA, and protein sequences. iLearnPlus which was built in 2021 is also a machine-learning platform. Some of its characteristics are:

1. Has graphical and web-based user interface.

2. Can run on multiple operating systems

3. Supports four formats for saving the calculated features, including LIBSVM, CSV, TSV, and WEKA

2.3.13 MathFeature

MathFeature [47], introduced in 2022 includes 37 features for biological sequences. 20 of them are based on mathematical approaches and are not available in other feature extraction packages. The other 17 features can be found in other approaches and are called conventional descriptors. Some of MathFeature's advantages are:

1. Suitable for RNA, DNA, and protein sequences

2. Contains GUI and web server

2.3.14 Feature extraction packages overview

Table 2.16 lists the 13 different Python-based programs which can extract features from sequences.

Package	**Number of features**	**Year**	**Ref**	**Used sequence**
propy	13	2013	[35]	Protein
PyDPI	13	2013	[36]	Protein
SPiCE	17	2014	[37]	Protein
PseKNC-General	11	2014	[38]	RNA and DNA
ProFET	24	2015	[39]	Protein
repDNA	15	2015	[40]	DNA and RNA
POSSUM	21	2017	[41]	Protein
iFeature	53	2018	[42]	Protein
PyBioMed	28	2018	[43]	DNA and Protein
BioSeq-Analysis	56	2019	[44]	RNA and DNA and Protein

Package	**Number of features**	**Year**	**Ref**	**Used sequence**
PyFeat	22	2019	[45]	RNA and DNA and Protein
iLearnPlus	117	2021	[46]	RNA and DNA and Protein
MathFeature	37	2022	[47]	RNA and DNA and Protein

Table 2.16: Feature extraction methods

For this project, we decided to use iLearnPlus as it is one of the most recent feature extraction programs and it is the one with the largest number of available feature sets.

2.4 Feature Selection Methods

Here, we describe the two methods we used for feature selection.

2.4.1 ANOVA

Analysis of variance, ANOVA, [48] determines each number in a variable's range and specifies how far each number is from the mean. The statistical technique called analysis of variance is used to determine whether the means of two or more groups differ significantly from one another. A feature's variance tells us how much it affects the response (target) variable. Low variance indicates that this feature has no effect on our target and vice-versa.

2.4.2 Mutual Information

Mutual Information (MI) [48] is a non-negative value, which measures the dependency between the variables. It is equal to zero if and only if two random variables are independent. Higher values mean higher dependency.

Chapter 3

Methodology

This project is to build a classifier for predicting sRNA-protein interactions in bacteria using only bacterial sRNA and protein sequence features. The classifier can only have two outputs, 0 or 1. 0 means sRNA and protein do not interact with each other, and 1 means sRNA and protein will interact with each other.

Regarding features, we only considered sequence-derived features. Structural features were not used since structures for all protein and sRNA sequences are not available. For the training data set, we collected experimentally validated pairs from several published studies.

3.1 Data Collection

Interacting pairs that were validated in experimental or wet-lab methods were used as positive examples. This means we have gathered sRNA-protein pairs which were defined to interact with each other as positive samples. In order to have sRNA sequences, after reviewing more than 200 papers, we used the studies which are listed in Table 3.1. It is good to mention that the given criteria are taken from their papers.

Organism (Strain)	**Criteria used to identify interacting pairs taken from original papers**	**Genome annotation number**	**Protein**	**Sequencing method(s)**	**# sRNA**	**Ref**
Escherichia coli (K-12)	At least 10 chimeric fragments	U00096.3	Hfq	RIL-seq	25	[49]
Escherichia coli (K-12)	Adjusted P-value ≤ 0.05	U00096.3	Hfq	co-ip and deep-sequencing	21	[50]
Salmonella enterica (LT2)	Enrichment factor ≥ 10	AE006468.2	Hfq	co-ip and deep-sequencing	25	[51]

Organism (Strain)	**Criteria used to identify interacting pairs taken from original papers**	**Genome annotation number**	**Protein**	**Sequencing method(s)**	**# sRNA**	**Ref**
Escherichia coli (Sakai)	Adjusted P-value ≤ 0.05	BA000007.2	Hfq	CRAC	27	[52]
Salmonella enterica (SL1344)	Correlations between the northern blot signals of sRNAs and their relative coverage in the cDNA libraries	FQ312003.1	Hfq	co-ip and deep-sequencing	63	[53]
Escherichia coli (MC4100)	wt IP/ wt total < 2	HG738867.1	Hfq	Tiling array of co-ip samples	34	[54]
Sinorhizobium meliloti (1021)	At least 30 strand-specific reads	AL591985.1	Hfq	co-ip and deep sequencing	94	[55]

Organism (Strain)	**Criteria used to identify interacting pairs taken from original papers**	**Genome annotation number**	**Protein**	**Sequencing method(s)**	**# sRNA**	**Ref**
Salmonella enterica (SL1344)	FDR-adjusted P-value < 0.05	FQ312003.1	proQ	ip and deep sequencing	108	[56]
Salmonella enterica (SL1344)	Enrichment factors of log2 fold change 2.0 with adjusted P-value < 0.05	FQ312003.1	proQ	co-ip and deep sequencing	61	[57]
Salmonella enterica (SL1344)	Enrichment factors of log2 fold change 2.0 with adjusted P-value < 0.05	FQ312003.1	FinO	co-ip and deep sequencing	6	[57]
Pseudomonas aeruginosa (PAO1)	Manual curation of sRNAs from size selection sRNA-seq	AE004091.2	Hfq	CLIP-seq	108	[57]

Organism (Strain)	**Criteria used to identify interacting pairs taken from original papers**	**Genome annotation number**	**Protein**	**Sequencing method(s)**	**# sRNA**	**Ref**
Escherichia coli (K-12)	IP enrichment ≥ 15	U00096.3	proQ	RIL-seq	63	[58]
Escherichia coli (K-12)	IP enrichment ≥ 15	U00096.3	Hfq	RIL-seq	105	[58]
Salmonella enterica (SL1344)	Adjusted P-value < 10E-4	FQ312003.1	CsrA	CLIP-seq	27	[59]
Salmonella enterica (SL1344)	Adjusted P-value < 10E-4	FQ312003.1	Hfq	CLIP-seq	126	[59]
Yersinia pestis biovar Microtus (91001)	Adjusted P-value < 0.001	AE01042.1	Hfq	CLIP-seq	456	[60]

Organism (Strain)	**Criteria used to identify interacting pairs taken from original papers**	**Genome annotation number**	**Protein**	**Sequencing method(s)**	**# sRNA**	**Ref**
Agrobacterium tumefaciens (C58)	ncRNAs enriched in Hfq^{3xFlag}	AE007869.2	Hfq	RIL-seq	113	[61]
Agrobacterium tumefaciens (C58)	ncRNAs enriched in Hfq^{3xFlag}	AE007870.2	Hfq	RIL-seq	49	[61]
Salmonella enterica (SL1344)	log2 fold change ≥ 2.0 and Adjusted P-value ≤ 0.05	FQ312003.1	proQ	RIP-seq	24	[62]
Bacillus subtilis (168)	RPKM > 2	AL009126.3	Hfq	co-ip and deep sequencing	22	[63]

Organism (Strain)	**Criteria used to identify interacting pairs taken from original papers**	**Genome annotation number**	**Protein**	**Sequencing method(s)**	**# sRNA**	**Ref**
Neisseria meningitidis (8013)	log2 fold change ≥ 2 and Adjusted p-value < 0.05	FM999788.1	proQ	CLIP-seq	16	[64]
Brucella suis (1330)	More than 20 cDNA reads	AE014291.4	Hfq	co-ip and deep sequencing	18	[65]
Brucella suis (1330)	More than 20 cDNA reads	AE014292.2	Hfq	co-ip and deep sequencing	15	[65]
Rhodobacter sphaeroides (2.4.1)	Enrichment factor > 1	CP000143.2	Hfq	RNA-seq	20	[66]
Erwinia amylovora (Ea1189)	0 < Hfq or 0 < Ea1189	FN666575.1	Hfq	RNA-seq	38	[67]

Organism (Strain)	**Criteria used to identify interacting pairs taken from original papers**	**Genome annotation number**	**Protein**	**Sequencing method(s)**	**# sRNA**	**Ref**
Clostridioides difficile (630)	FDR-adjusted P-value ≤ 0.1 and log2 fold change ≥ 2	CP010905.2	Hfq	RIP-seq	26	[68]

Table 3.1: Studies used to collect sRNA-protein interacting pairs

The criteria for considering whether a sRNA and a RBP interact when analyzing sequencing data, that I listed in the Table 3.1, were mentioned in the original papers. The criteria vary across the various publications and are not directly comparable. However, our purpose is to obtain examples of interacting sRNA-RBP pairs.

In Table 3.2, we provide a brief explanation of the sequencing methods listed in Table 3.1.

Method name	**Explanation**	**Reference**
RIL-seq	RNA interaction by ligation and sequencing.	[49]
Deep sequencing	High-throughput sequencing.	[50]
CRAC	UV-induced RNA-protein crosslinking and analysis of cDNA by high-throughput sequencing.	[52]
Co-ip	Identify physiologically relevant protein-protein interactions by using target protein-specific antibodies to indirectly capture proteins that are bound to a specific target protein.	[69]
ip	A small-scale affinity purification of antigens using a specific antibody that is immobilized to a solid support such as magnetic particles or agarose resin.	[69]
CLIP-seq	Cross-linking immunoprecipitation followed by deep sequencing.	[70]
RNA-seq	Deep sequencing of cDNA libraries.	[70]
RIP-seq	RNA immunoprecipitation sequencing.	[68]

Table 3.2: Sequencing-based methods used in the studies listed in Table 3.1

We collected the genomic coordinates of 1559 sRNAs (BED file) to obtain their sequences from the corresponding genome (fasta file). The genomic coordinates were specified in the corresponding publication. To do this, we used bedtools getfasta [71] as follows:

```
$ bedtools getfasta [OPTIONS] -fi <FASTA> -bed <BED/GFF/VCF>
```

With the following arguments:

- We set -s option to force strandedness
- Input FASTA file with whole genome in fasta format /GFF/VCF file with sRNA coordinates

For protein sequences, UniProt [72] was used. We retrieved 16 different sequences for our proteins (Hfq, CsrA, ProQ, and FinO) in different bacteria.

3.1.1 Interacting Pairs

1559 sRNA sequences with their interacting proteins were collected as a positive data set. Table 3.3 shows the properties of the training data set. Also, we used complete sequences instead of only binding domains.

Number of unique protein sequences	**Number of unique sRNA sequences**	**Number of interacting pairs**	**Number of non-interacting pairs**
16	1559	1559	1559

Table 3.3: Properties of the training data set

Note that in Table 3.1 there are 26 proteins listed; however, out of these, there are only 16 unique sequences.

3.1.2 Non-interacting Pairs

To generate non-interacting pairs to train our model, we randomly selected a protein sequence (one of the 16 proteins) from our training data set and genomic coordinates from the corresponding bacterial genome. This process was repeated until the number of non-interacting pairs was equal to the number of interacting pairs.

We used bedtools shuffle [73] to obtain random genomic locations of the same length as the sRNAs on our training data. Then bedtools getfasta [69] was used to retrieve the sequences corresponding to the random genomic locations.

1. bedtools shuffle:

   ```
   bedtools shuffle [OPTIONS] -i <BED/GFF/VCF> -g <GENOME>
   ```

   with the following options:

   - 50% was chosen as a maximum overlap of random genomic locations with actual sRNAs to ensure that the sequence of the non-interacting pairs is substantially different from actual sRNAs.
   - BED/GFF/VCF refers to an input bed file with sRNA coordinates
   - GENOME refers to a text file with the length of the corresponding bacterial genome
   - Output file refers to another bed file with the random genomic locations that we get at the end

2. bedtools getfasta:

   ```
   bedtools getfasta [OPTIONS] -fi <FASTA> -bed <BED/GFF/VCF>
   ```

   With the following options:

   - We set the -s option to force strandedness.
   - Input FASTA file with the whole genome
   - BED/GFF/VCF file with coordinates of negative instances

3.2 Testing Data Sets

In order to test our model for general RNA-protein interaction prediction and compare it with other programs, we used the four data sets listed in Table 3.4.

Data set	**Number of RNAs**	**Number of proteins**	**Number of + pairs**	**Number of - pairs**	**Ref**
RPI369	331	623	369	369	[8]
RPI1807	646	868	652	221	[11]
RPI488	13	155	43	47	[11]
NPInter v2.0	513	448	1943	1943	[11]

Table 3.4: Testing data sets

3.3 Feature Extraction

For extracting features (attributes) from sequences, we used the iLearnPlus [46] Python package because:

- It can be used for protein and sRNA sequences
- It is a recently published package (2021)
- It is able to extract multiple features

Table 3.5 shows the different sRNA feature sets which were extracted using iLearnPlus.

Feature set	**Description**	**Reference**
NAC (Nucleic Acid Composition)	NAC calculates the frequency of each nucleic acid type in a nucleotide sequence.	[46]
DNC (Dinucleotide Composition)	Frequency of each nucleotide pair in a sequence.	[46]
TNC (Trinucleotide Composition	Frequency of each nucleotide trio in a sequence.	[46]
PseEIIP (Electron-ion Interaction Pseudopotentials of Trinucleotide)	calculates the pseudo-electron-ion interaction for each trinucleotide sequence. It creates a feature vector for each sequence. The vector contains a value for each trinucleotide. The value is computed by multiplying the aggregate value of electron-ion interaction of each trinucleotide.	[74]

Feature set	**Description**	**Reference**
ASDC (Adaptive Skip Dinucleotide Composition)	This descriptor considers the correlation information present not only between adjacent residues but also between intervening residues.	[46]
MMI	Multivariate mutual information with 2-mer and 3-mer DNA/RNA sequence.	[46]
Z-curve-9bit (The Z curve parameters for frequencies of phase-specific mononucleotides)	The frequencies of bases A, C, G, and T occurring in an open reading frame or a fragment of DNA sequence. They are in fact the frequencies of bases at the 1st, 2nd and 3rd codon positions.	[46]
CKSNAP (Composition of K-spaced Nucleic Acid Pairs)	Calculates the frequency of nucleic acid pairs separated by any 3 nucleic acid.	[46]

Feature set	**Description**	**Reference**
RCKmer (Reverse Compliment Kmer)	A variant of kmer-descriptor, in which the kmers are not expected to be strand-specific.	[46]

Table 3.5: Extracted sRNA sequence feature sets with iLearnPlus Python package

Extracted features from protein sequences using iLearnPlus are listed in Table 3.6.

Feature set	**Description**	**Reference**
AAC (Amino Acid Composition)	Calculates the frequency of each amino acid type in a protein or peptide sequence.	[46]
DPC (Di-Peptide Composition)	Computes the frequency of two amino acids. A protein sequence can be represented by a 400-dimensional vector.	[46]
DDE	Computes the dipeptide deviation from the expected mean value.	[46]
GAAC (Grouped Amino Acid Composition)	The 20 amino acid types are categorized into five classes according to their physicochemical properties and GAAC is the frequency of each amino acid group.	[46]
GDPC (Grouped Dipeptide Composition)	Combination between DPC and GAAC.	[75]
GTPC (Grouped Tripeptide Composition)	Combination between TPC and GAAC. TPC computes the frequency of three amino acids which can be represented by a 8000-dimensional vector.	[75]

Feature set	**Description**	**Reference**
CTDC (Composition)	The composition consists of three values: the global compositions (percentage) of polar, neutral, and hydrophobic residues of the protein. This feature set can be shown by a 39-dimensional vector.	[46]
CTDT (Transition)	The transition also consists of three values: the global compositions (percentage) of polar, neutral and hydrophobic residues of the protein. This feature set can be shown by a 39-dimensional vector.	[46]
CTDD (Distribution)	The distribution consists of five values for each of the three groups (polar, neutral and hydrophobic), namely the corresponding fraction of the entire sequence.	[46]
ASDC	It considers the correlation information present not only between adjacent residues but also between intervening residues. This function calculates the frequency of pair amino acids omitting gaps between them. Then this function normalizes each value by dividing each frequency by the sum of all frequencies.	[76]

Table 3.6: Extracted protein sequence feature sets using iLearnPlus Python package

As a result of feature extraction, we extracted 9 feature sets (331 features) from sRNA sequences and 10 feature sets (1648 features) from protein sequences. Our feature table has 3118 (number of interacting and non-interacting sequences) rows and 1979 (number of sRNA and protein features) columns.

Our first step to reduce the number of similar features was to remove them. We removed less important features for reducing noise and enhancing model performance. To do this, we calculated pair-wise Spearman correlation among all feature sets and removed those with a correlation value greater than 0.90. We used 0.9 as a value because we wanted to remove highly redundant features. That is features that provide practically the same information. After removing redundant features we had 182 sRNA features and 743 protein features. Table 3.7 shows the number of remaining features after filtering based on the pair-wise Spearman correlation values between features.

Feature sets	**Feature type**	**Number of features before correlation**	**Number of features after correlation**
NAC	RNA/DNA	4	4
DNC	RNA/DNA	16	16
TNC	RNA/DNA	64	63
PseEIIP	RNA/DNA	64	0
ASDC	RNA/DNA	16	12
MMI	RNA/DNA	30	30
Z-curve-9bit	RNA/DNA	9	9
CKSNAP	RNA/DNA	64	48
RCKmer	RNA/DNA	64	0
AAC	Protein	20	20

Feature sets	**Feature type**	**Number of features before correlation**	**Number of features after correlation**
DPC	Protein	400	254
DDE	Protein	400	63
GAAC	Protein	5	3
GDPC	Protein	25	16
GTPC	Protein	125	72
CTDC	Protein	39	15
CTDT	Protein	39	22
CTDD	Protein	195	94
ASDC	Protein	400	184

Table 3.7: Number of sequences filtered by removing features with a pair-wise Spearman correlation value greater than 0.9

After the removal of redundant features, three of the feature sets were completely removed. This happened because all the features in those feature sets were similar to each other. I calculated the correlation within each of the feature sets and removed all the features if and only if they had a correlation greater than 0.9 in the same feature set.

3.4 Feature Selection

We implemented ANOVA and Mutual Information (MI) methods for feature selection [48]. The data set that we used for feature selection contains 3118 rows and 925 columns after removing redundant features with a Spearman correlation coefficient greater than 0.9.

Several ANOVA and Mutual Information thresholds were considered. Each threshold can only consider a subset of features. In Figures 3.1 and 3.2, the horizontal axe shows the features and the vertical axe shows their corresponding scores. We were aiming to select those features with a high score because these features exhibit significant differences in means across different groups or categories of the target variable. To do this, the following thresholds were used:

(a) ANOVA100

(b) ANOVA200

(c) MI0.2

(d) MI0.4

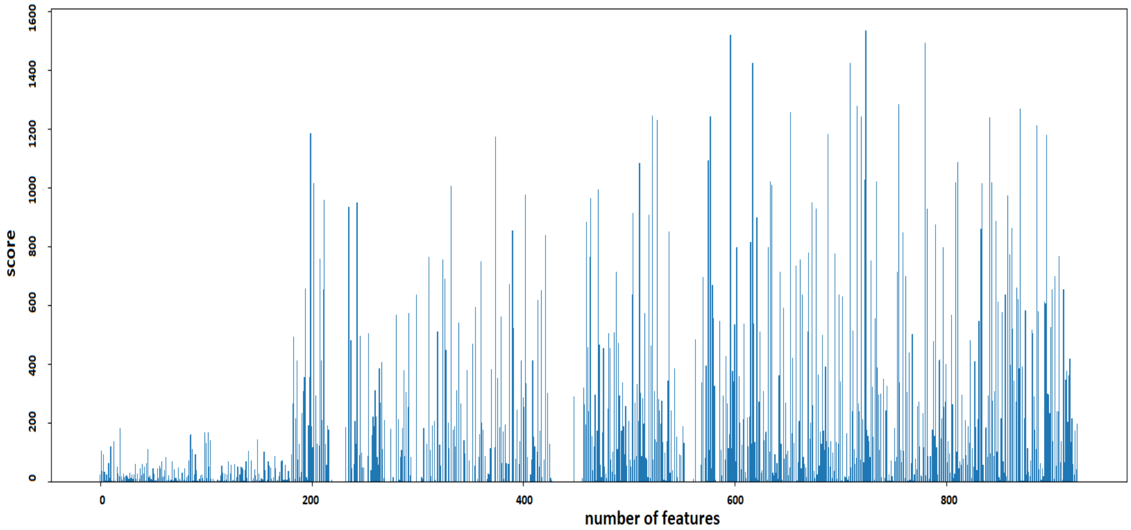


Figure 3.1: ANOVA feature selection result in our data set

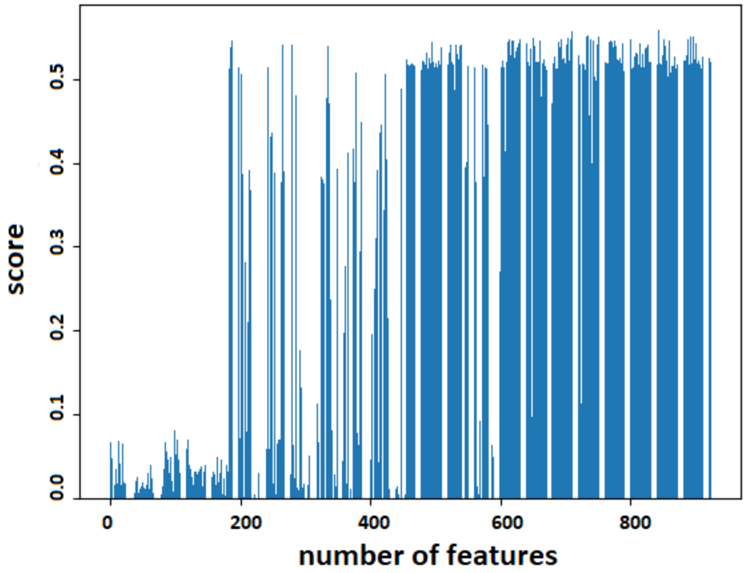


Figure 3.2: Mutual Information feature selection result in our data set

For each of the machine learning methods, different hyperparameters were tested to find the values which optimize classification performance. Table 3.8 shows the hyperparameters used to construct the models for each machine-learning method. These hyperparameters were taken from grid-search and further used to evaluate classifiers based on a 10-fold CV.

Classifier	**Hyperparameters**
Decision tree	criterion:entropy, max-depth:14
xGBoost	gamma:0.01, learning-rate:0.1, max-depth:6, n-estimators:500
Random forest	criterion:gini, max-depth:14, max-features:auto, n-estimators:20

Table 3.8: Classifiers hyperparameters

Tables 3.9, 3.10, and 3.11 show the different hyperparameter values used for grid-search in DT, XGBoost, and RF machine-learning methods, respectively.

Hyperparameter	**Variables**
Criterion	gini - entropy
Max-depth	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - None

Table 3.9: DT hyperparameter values explored with grid-search cross-validation

Hyperparameter	**Variables**
N-estimators	10 - 20 - 30
Max-depth	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - None
Gamma	0.1 - 0.01
Learning-rate	0.001 - 0.01 - 0.1 - 1

Table 3.10: XGBoost hyperparameter values explored with grid-search cross-validation

Hyperparameter	**Variables**
N-estimators	1 - 5 - 10 - 15
Max-features	auto - sqrt - log2
Criterion	gini - entropy
Max-depth	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - None

Table 3.11: RF hyperparameter values explored with grid-search cross-validation

Figure 3.3 represents the percentage and number of sRNA and protein features in the ANOVA100 feature set, which was used for the training. Surprisingly, most of the features found informative by ANOVA are extracted from the protein sequence.

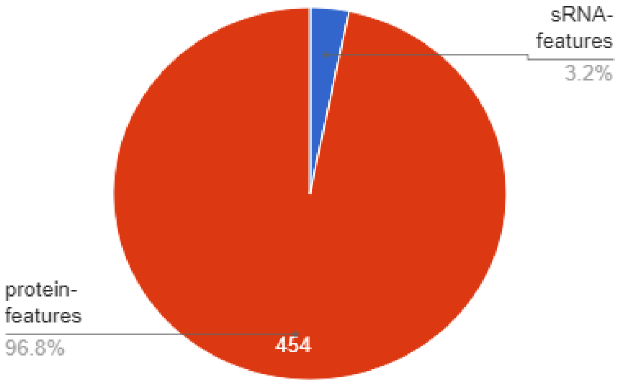


Figure 3.3: ANOVA100 features grouped by their types (sRNA or protein)

Table 3.12 shows the number of features in ANOVA100.

Feature set	**Number of features**	**Feature type**
NAC	1	RNA/DNA
DNC	3	RNA/DNA
TNC	1	RNA/DNA
ASDC	3	RNA/DNA
MMI	4	RNA/DNA
CKSNAP	3	RNA/DNA
AAC	14	Protein
DPC	98	Protein
DDE	48	Protein
GAAC	2	Protein
GDPC	14	Protein
CTPC	36	Protein
CTDC	12	Protein
CTDT	19	Protein
CTDD	67	Protein
ASDC	144	Protein

Table 3.12: ANOVA100 features

Chapter 4

Results and Discussion

In this chapter, we present performance metrics for various thresholds for feature selection, and three machine-learning methods. We present the results of a comparative assessment of our best model ProNA's predictive performance with that of other recent RNA-protein prediction programs.

4.1 Feature Selection

Table 4.1 shows the number of features selected with two different thresholds for ANOVA and Mutual Information.

Method	**Threshold**	**Selected features**
ANOVA	100	469
ANOVA	200	394
Mutual Information	0.2	531
Mutual Information	0.4	476

Table 4.1: Number of features selected with two feature selection methods at different threshold settings

Table 4.2 shows the 10-fold CV (cross-validation) accuracy obtained from three different machine-learning methods with four different feature sets. Based on Table 4.2, we chose ANOVA100 with XGBoost as it has better accuracy with fewer features in comparison with the other 3 feature sets. Nevertheless, all machine learning and feature set combinations have very similar performance. This is likely due to the fact that the feature sets have a large intersection between each other (Figure 4.1) and all machine learning methods used are tree-based approaches.

Method	**ANOVA100** **(3118*469)**	**ANOVA200** **(3118*394)**	**MI0.2** **(3118*531)**	**MI0.4** **(3118*476)**
decision tree (grid-search)	0.934	0.938	0.936	0.936
decision tree (CV)	0.903±0.03	0.938 ±0.03	0.938 ±0.03	0.938 ±0.03
XGBoost (grid-search)	0.943	0.938	0.936	0.936
XGBoost (CV)	0.948 ±0.02	0.938 ±0.03	0.938 ±0.03	0.938 ±0.03
Random forest (grid-search)	0.936	0.938	0.936	0.936
Random forest (CV)	0.94 ±0.02	0.938 ±0.03	0.938 ±0.03	0.938 ±0.03

Table 4.2: Accuracy obtained using different feature sets. Between parenthesis, the first number shows the number of sequences and the second number shows the number of features.

In Figure 4.1, common features between 4 feature sets are shown:

(a) ANOVA100 and MI0.2

(b) ANOVA200 and MI0.4

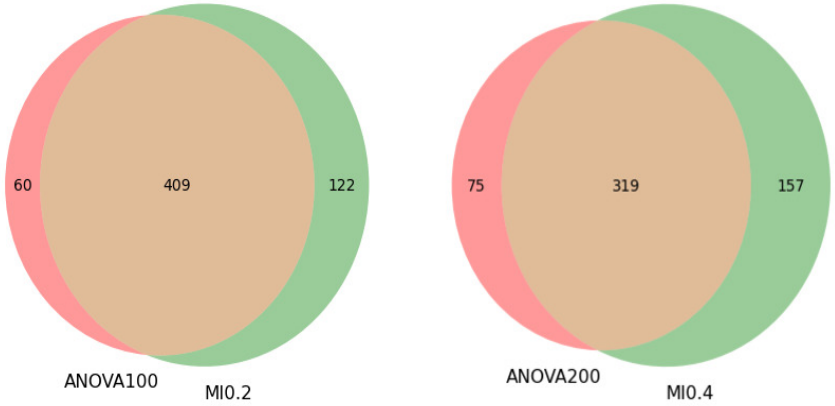


Figure 4.1: Similarity between different feature sets

Figure 4.1 shows the number of features selected by ANOVA and mutual information. All the features in common are protein features which suggest protein features have a strong signal for this task. The reason we selected these two groups is their number of features.

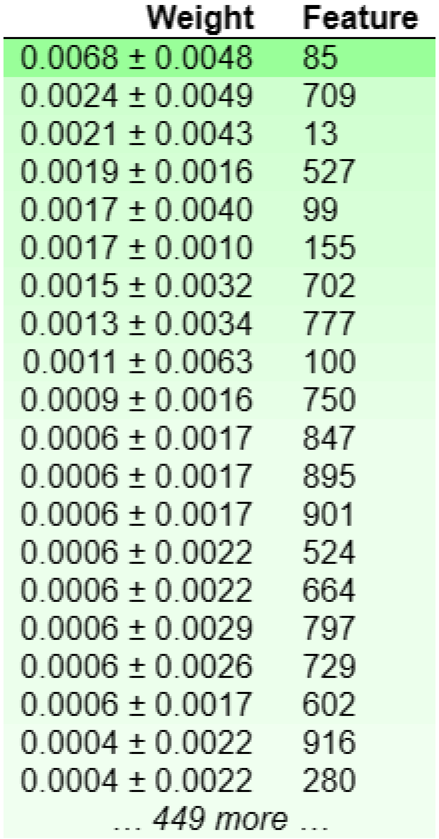
4.2 Feature Importance

After feature selection, based on Table 4.2, we tried feature permutation with three classifiers (XGBoost, random forest, and decision tree) to determine the importance of each feature in a machine-learning model. Feature permutation evaluates the importance of features in a data set by randomly shuffling the values of a single feature in the data set and then re-evaluating the model's performance.

The five most important features in the Random Forest classifier are shown in Table 4.3 and Figure 4.2.

Feature	**Feature set**	**Feature type**
AT	ASDC	DNA/RNA
Polarity.2. residue25	CTDD	Protein
GG	DNC	DNA/RNA
postivecharger. alphaticr	GDPC	Protein
CC	MMI	DNA/RNA

Table 4.3: The five most important features in the RF classifier



Weight	Feature
0.0068 ± 0.0048	85
0.0024 ± 0.0049	709
0.0021 ± 0.0043	13
0.0019 ± 0.0016	527
0.0017 ± 0.0040	99
0.0017 ± 0.0010	155
0.0015 ± 0.0032	702
0.0013 ± 0.0034	777
0.0011 ± 0.0063	100
0.0009 ± 0.0016	750
0.0006 ± 0.0017	847
0.0006 ± 0.0017	895
0.0006 ± 0.0017	901
0.0006 ± 0.0022	524
0.0006 ± 0.0022	664
0.0006 ± 0.0029	797
0.0006 ± 0.0026	729
0.0006 ± 0.0017	602
0.0004 ± 0.0022	916
0.0004 ± 0.0022	280

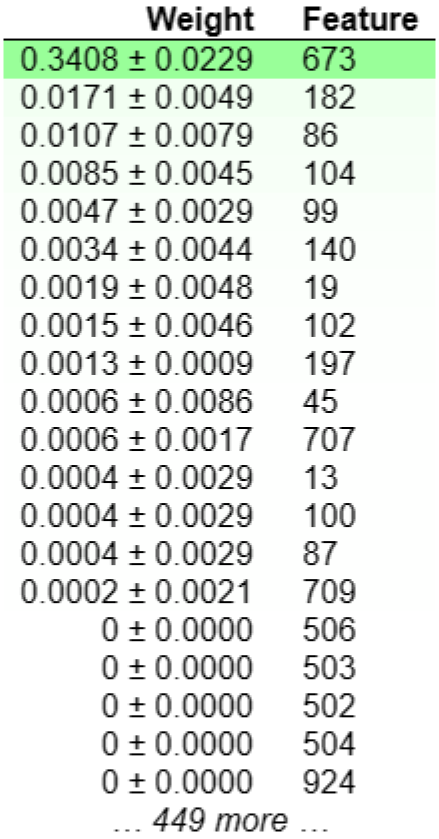
… 449 more …

Figure 4.2: RF classifier feature permutation result on ANOVA100 feature set, sorted by most important features based on their weights

Table 4.4 and Figure 4.3 depicts the five most important features in the XGBoost classifier.

Feature	**Feature set**	**Feature type**
TA	ASDC	Protein
AT.gap0	CKSNAP	DNA/RNA
CA	ASDC	DNA/RNA
TT	MMI	DNA/RNA
CC	MMI	DNA/RNA

Table 4.4: The five most important features in the XGBoost classifier



Weight	Feature
0.3408 ± 0.0229	673
0.0171 ± 0.0049	182
0.0107 ± 0.0079	86
0.0085 ± 0.0045	104
0.0047 ± 0.0029	99
0.0034 ± 0.0044	140
0.0019 ± 0.0048	19
0.0015 ± 0.0046	102
0.0013 ± 0.0009	197
0.0006 ± 0.0086	45
0.0006 ± 0.0017	707
0.0004 ± 0.0029	13
0.0004 ± 0.0029	100
0.0004 ± 0.0029	87
0.0002 ± 0.0021	709
0 ± 0.0000	506
0 ± 0.0000	503
0 ± 0.0000	502
0 ± 0.0000	504
0 ± 0.0000	924

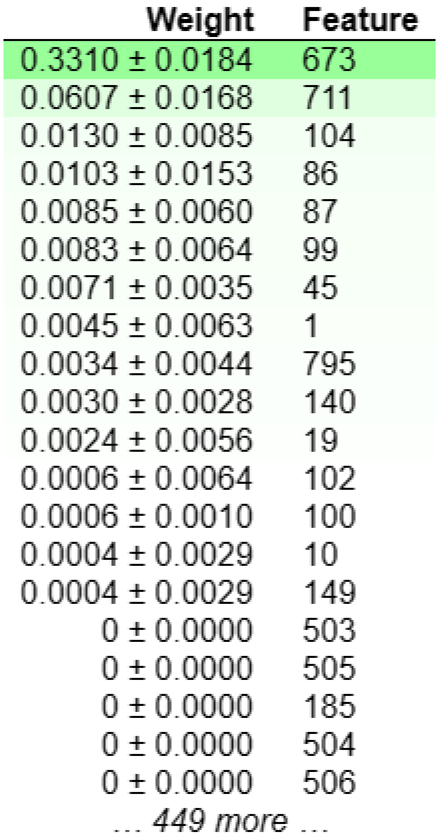
… 449 more …

Figure 4.3: XGBoost classifier feature permutation result on ANOVA100 feature set, sorted by most important features based on their weights

The five most important features in the Decision Tree classifier are displayed in Table 4.5 and Figure 4.4.

Feature	Feature set	Feature type
hydrophobicity-PONP930101.1.residue25	CTDD	Protein
hydrophobicity-ARGP820101.1.residue25	CTDD	Protein
TT	MMI	DNA/RNA
CA	ASDC	DNA/RNA
CC	ASDC	DNA/RNA

Table 4.5: The five most important features in the DT classifier



Weight	Feature
0.3310 ± 0.0184	673
0.0607 ± 0.0168	711
0.0130 ± 0.0085	104
0.0103 ± 0.0153	86
0.0085 ± 0.0060	87
0.0083 ± 0.0064	99
0.0071 ± 0.0035	45
0.0045 ± 0.0063	1
0.0034 ± 0.0044	795
0.0030 ± 0.0028	140
0.0024 ± 0.0056	19
0.0006 ± 0.0064	102
0.0006 ± 0.0010	100
0.0004 ± 0.0029	10
0.0004 ± 0.0029	149
0 ± 0.0000	503
0 ± 0.0000	505
0 ± 0.0000	185
0 ± 0.0000	504
0 ± 0.0000	506

… 449 more …

Figure 4.4: DT classifier feature permutation result on the ANOVA100 feature set, sorted by most important features based on their weights

The di-nucleotide CC was among the five most important features for all three classifiers. Additionally, TT and CA were both among the five most important features for DT and XGBoost. The most important protein features were unique for each classifier. Each classifier has a different number of sRNA and protein features. From Figs. 4.2 - 4.4, one can see that the weights of the features vary among classifiers: For XGBoost and decision tree the most important feature weights substantially more than all other features; while for RF all features have comparable weights. It is also noticeable that even though RNA features comprised the minority of the features, they are found to be among the most relevant by the classifiers.

4.3 10-Fold CV

To determine the best classifier (machine learning model), we evaluated them with our feature set (ANOVA100) and determined that XGBoost has the best accuracy. The following evaluation metrics were used for comparing the classifiers:

(a) AUC-ROC: AUC-ROC stands for Area Under the Receiver Operating Characteristic Curve. It is a metric used to evaluate the performance of binary classification models, which are models that predict one of two possible outcomes (usually represented as 0 and 1).

(b) AUC-PR: AUC-PR stands for Area Under the Precision-Recall Curve. Similar to AUC-ROC, it's a metric used to evaluate the performance of binary classification models, but it focuses on the precision-recall trade-off rather than the true positive rate and false positive rate.

(c) Accuracy: Accuracy is a common metric used to evaluate the performance of classification models. It measures the proportion of correctly predicted instances out of the total instances in the dataset. In other words, accuracy quantifies how well the model's predictions match the actual true labels.

(d) F1 Score: The F1 score is a metric commonly used in binary classification to provide a balance between precision and recall. It considers both false positives and false negatives and is particularly useful when the class distribution is imbalanced.

$$F1 = \frac{2 \cdot (\text{precision} \cdot \text{recall})}{\text{precision} + \text{recall}}$$

Table 4.6 shows the different performance metrics of three classifiers.

Metric	**RF**	**XGBoost**	**DT**
ROC AUC	0.986±0.004	0.991±0.003	0.929±0.010
PR AUC	0.988±0.004	0.991±0.003	0.944±0.010
Accuracy	0.939±0.010	0.947±0.013	0.902±0.015
F1 Score	0.937±0.010	0.946±0.013	0.924±0.013

Table 4.6: Performance metrics on three classifiers with ANOVA100 feature set

Before plotting the curves, understanding how to interpret each plot is essential. At each of the 10 folds:

- The ROC curve shows the trade-off between TPR and FPR. A perfect classifier would have TPR = 1 and FPR = 0, which means it correctly identifies all positive cases and makes no false positive errors. The worst classifier would have TPR = FPR = 0.5, which means it performs no better than random guessing.

- The area under the PR curve (AUC-PR) is a common metric used to quantify the overall performance of the classifier. A higher AUC-PR value (ranging from 0 to 1) indicates better model performance in terms of precision and recall.

Here, we show ROC and PR curves next to each other for all of the classifiers.

(a) Random forest classifier is shown in Figure 4.5. This method was evaluated with the following hyperparameters:

criterion:gini, max-depth:14, max-features:auto, n-estimators:20

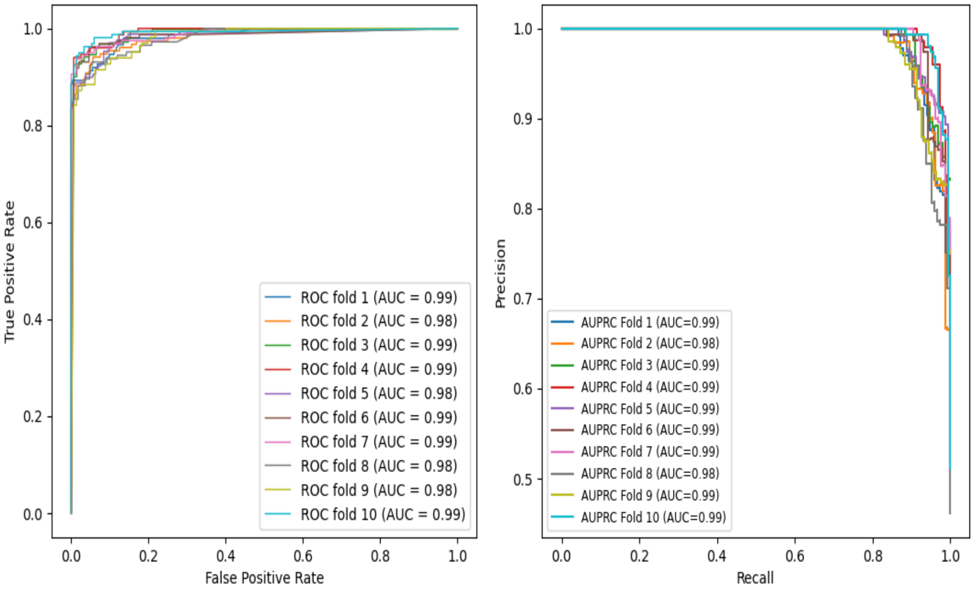


Figure 4.5: Random forest classifier evaluated with ROC and PR curves

(b) Figure 4.6 shows the XGBoost classifier. This method was evaluated with the following hyperparameters:

gamma:0.01, learning-rate:0.1, max-depth:6, n-estimators:500

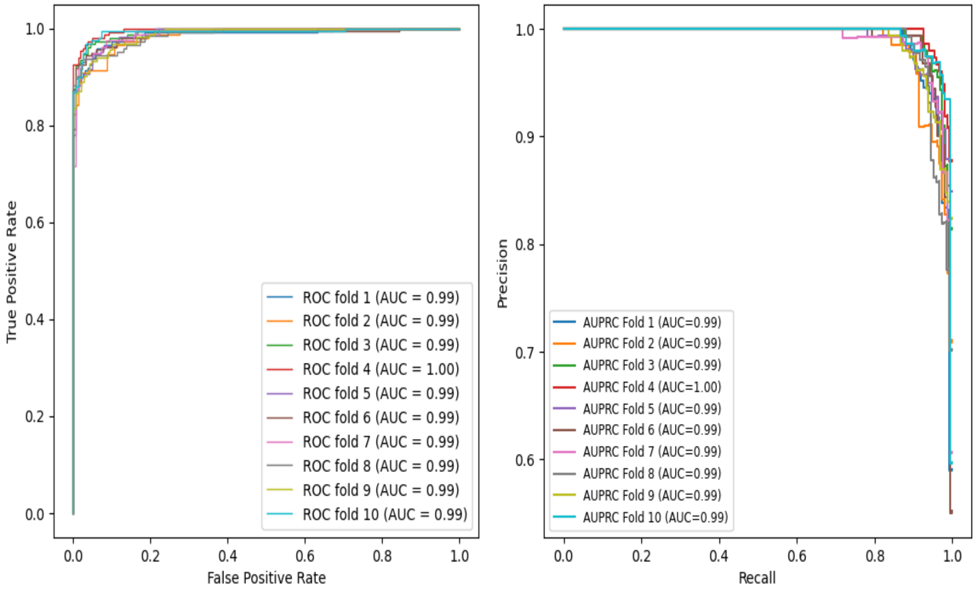


Figure 4.6: XGBoost classifier evaluated with ROC and PR curves

(c) Figure 4.7 depicts decision tree classifier. This method was evaluated with the following hyperparameters:

criterion: entropy, max-depth:14

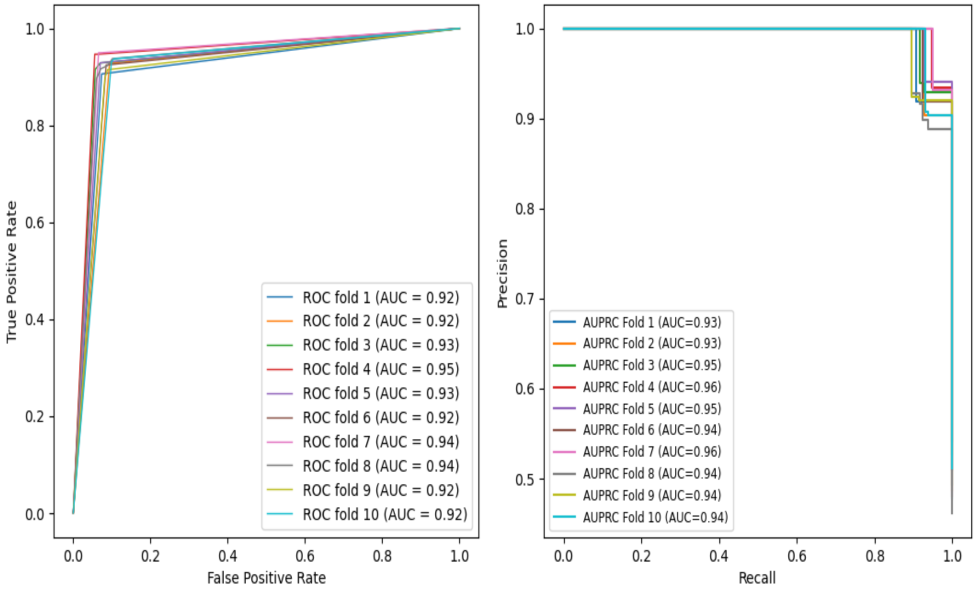


Figure 4.7: Decision tree classifier evaluated with ROC and PR curves

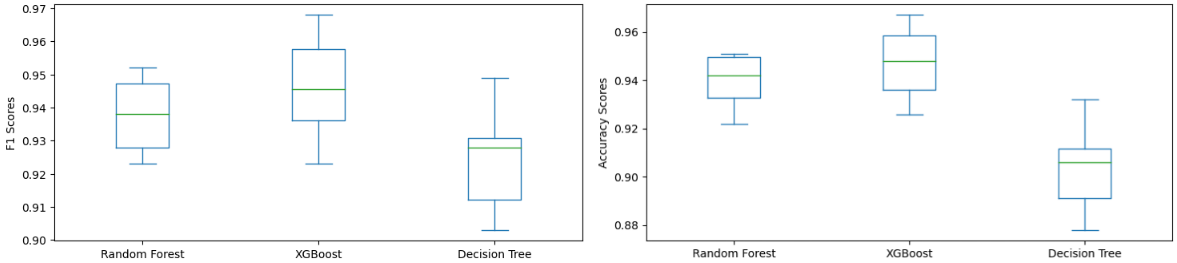


Figure 4.8: F1 and accuracy scores for 3 classifiers

XGBoost has the highest F1 scores and accuracy among the three methods. Although its performance is comparable to that of Random Forest. Based on Figs. 4.5 and 4.6, XGBoost seems to have slightly less variation (i.e., the CV-fold lines clustered more together) in cross-validation than Random Forest. Thus, our final model (called ProNA) was built using:

i. XGBoost classifier

ii. ANOVA100 feature set

4.4 Comparative Assessment

For comparing ProNA with other programs, four data sets frequently used by the approaches described in Section 2.1 were selected:

i. RPI369

ii. RPI488

iii. RPI1807

iv. NPInter v2.0

We used ProNA to estimate the probability of interaction for every sRNA-protein pair on these data sets and evaluated its predictive performance. We compared ProNA's performance with the performance of other programs. Tables 4.7, 4.8, 4.9, and 4.10 compare the performance of our model (ProNA) with other programs on RPI369, RPI488, RPI1807, and NPInter v2.0 data sets, respectively. Note that as some of the programs are no longer available or we were unable to run some of the programs, the performance metrics reported in these tables are as reported by the corresponding manuscript. However, we observed that performance metrics provided for a given program in another program's manuscript might differ from the performance reported in the original publication. This might be due to variations in the data sets used. For example, there are several versions of the NPInter data set. It is important to mention that different evaluation metrics were used for each data set. In other words, we used only common evaluation metrics between all the programs.

Programs	**Accuracy**	**Recall**
RPI-Pred	0.92	0.89
RPI-SE	0.88	0.83
RPiRLS	0.85	0.86
ProNA	0.84	0.93
RPiRLS-7G	0.79	0.87
DM-RPIs	0.79	0.83
RPISeq-RF	0.76	0.78
De novo-ENB	0.75	0.34
De novo-NB	0.74	0.36
RPITER	0.72	0.79
RPISeq-SVM	0.72	0.73
SAWRPI	0.71	0.75

Table 4.7: Results of various programs for RNA-protein prediction on RPI369 data set

Accuracy and Recall were two evaluation metrics in all the programs, Thus we used these two metrics and compare our program with other programs based on these two metrics.

Programs	**Accuracy**	**Precision**	**Recall**	**MCC**
ProNA	0.90	0.97	0.93	0.89
RPI-SAN	0.89	0.95	0.94	0.79
RPITER	0.89	0.94	0.83	0.79
RPI-SE	0.89	0.95	0.94	0.79
SAWRPI	0.89	0.93	0.84	0.79
EDLMFC	0.86	0.96	0.74	0.74

Table 4.8: Results of various programs for RNA-protein prediction on RPI488 data set

Programs	**Accuracy**	**Precision**	**Recall**	**MCC**
RPI-SAN	0.96	0.91	0.93	0.92
RPITER	0.96	0.95	0.98	0.93
RPI-SE	0.96	0.95	0.96	0.93
SAWRPI	0.96	0.96	0.98	0.93
EDLMFC	0.93	0.94	0.96	0.83
ProNA	0.88	0.95	0.91	0.85

Table 4.9: Results of various programs for RNA-protein prediction on RPI1807 data set

Programs	**Accuracy**	**Precision**	**Recall**	**MCC**	**Specificity**
RPITER	0.95	0.93	0.97	0.91	0.93
ProNA	0.92	0.96	0.97	0.97	0.82
EDLMFC	0.89	0.88	0.91	0.79	0.87
De novo-ENB	0.77	0.76	0.47	0.46	0.92
De novo-NB	0.74	0.73	0.35	0.37	0.93

Table 4.10: Results of various programs for RNA-protein prediction on NPInter v2.0 data set

For the NPInter v2.0 and RPI369 data sets, ProNA's performance is comparable to that of the other approaches. ProNA outperforms other approaches for data set RPI488 which contains lncRNA-protein interactions. ProNA's performance is the poorest for the RPI1807 data set which contains ncRNA-protein pairs. The results of ProNA are quite good considering that ProNA was trained on a completely different data set and some of the other programs were trained on these data sets and their reported performance is from a CV process. Additionally, RPITER, EDLMFC, and RPI-Pred use structures and sequences as input data, while ProNA only uses sequence-based features.

4.5 ROC and PR Curves

Figures 4.9, 4.10, 4.11, and 4.12 illustrate ProNA ROC and PR curves in RPI369, RPI488, RPI1807, and NPInter v2.0 data sets, respectively.

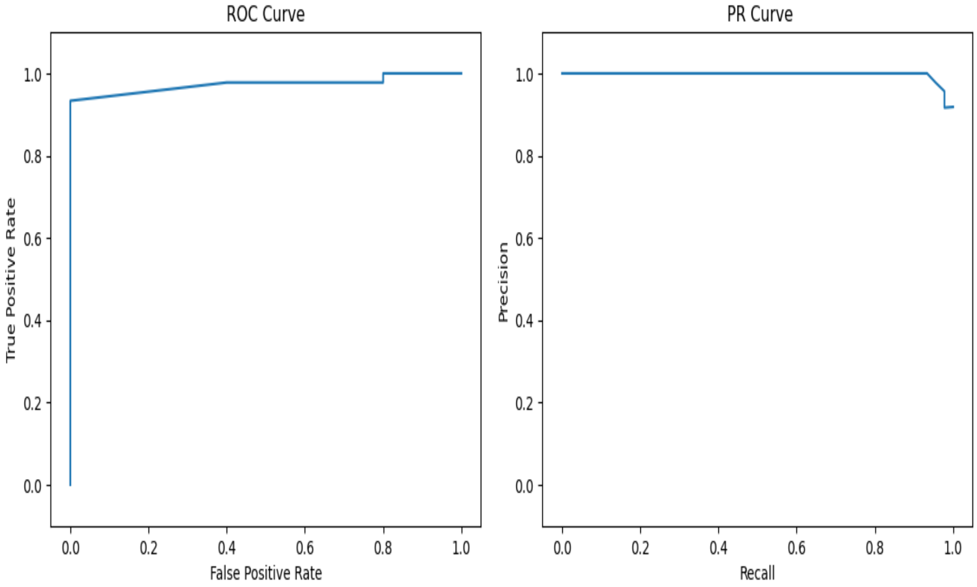


Figure 4.9: ProNA result in RPI369 data set

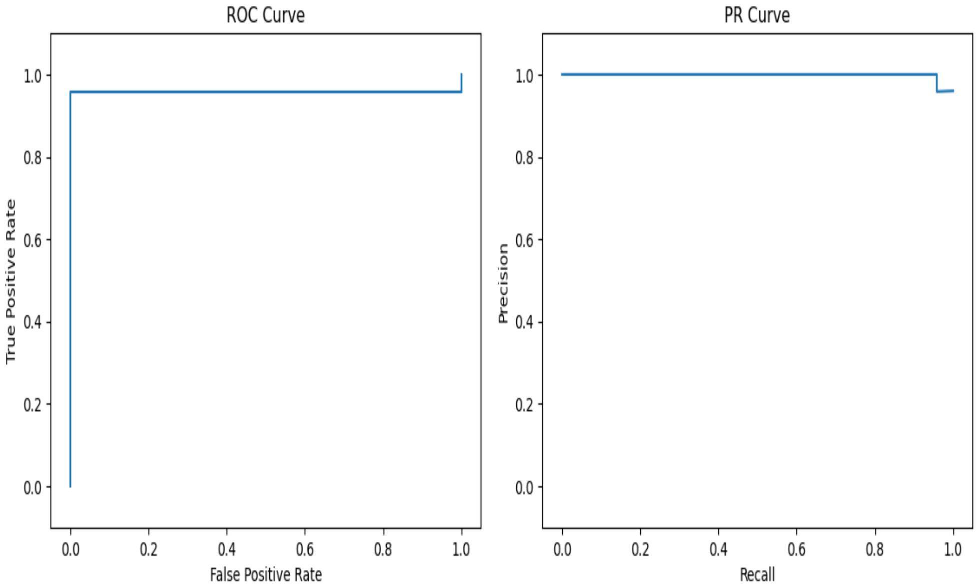


Figure 4.10: ProNA result in RPI488 data set

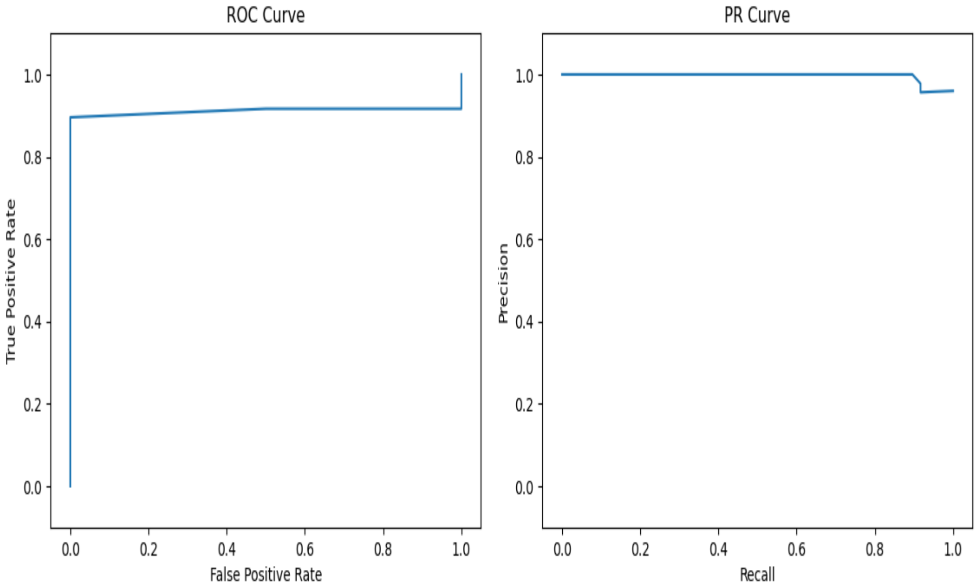


Figure 4.11: ProNA result in RPI1807 data set

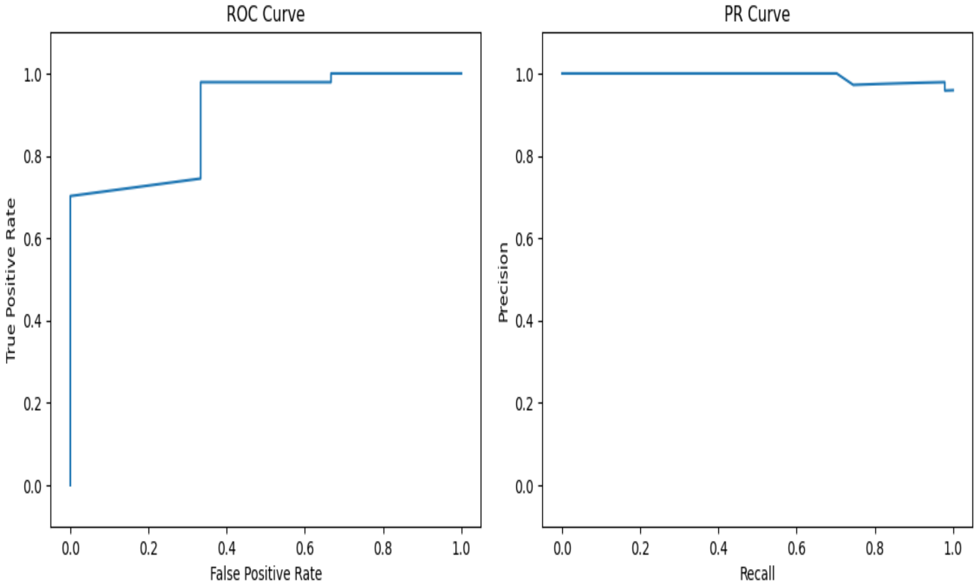


Figure 4.12: ProNA result in NPInter v2.0 data set

4.6 Assessment on an independent bacterial data set

Recently a small number of sRNA-protein interactions in the bacterium *Pasteurella multocida* have been experimentally determined by Gulliver et al. [77] and Marianne Megroz [78]. In total both studies identified two sRNAs interacting with the protein Hfq and six sRNAs interacting with ProQ. None of these sRNA-protein interactions were in any of the data sets previously mentioned.

We created a FASTA file with the sequences of 32 sRNAs and another with two protein sequences of *Pasteurella multocida* and calculated their probability of interaction using ProNA and RPISeq RF (available as a web application). Considering as predicted interactions all sRNA-protein pairs with a probability of interaction greater than 0.5, ProNA identified 2 out of 8 experimentally determined pairs and in total predicted 27 interacting pairs; while RPISeq RF identified 4 out of 8 experimentally determined pairs and in total predicted 49 interacting pairs. If we assume all other sRNA-protein pairs are not interacting then ProNA has a precision of 7.4%, while RPISeq RF has a precision of 8.2%. Thus, although ProNA has a comparable performance with that of RPISeq RF, its performance does not support our initial hypothesis that a classifier trained on bacterial sRNA and protein sequences would outperform other classifiers trained on a general RPI data set. Additionally, our results show that more work is needed to improve the predictive performance of current programs for predicting bacterial sRNA-protein interactions.

Chapter 5

Conclusion

In this thesis, I build a machine-learning method for sRNA-protein interaction predictions in bacteria using only sequence-based features. Our model, ProNA, takes two input FASTA files, one with the protein sequences and one with the sRNA sequences, and returns the probabilities of non-interaction and interaction between all possible sRNA-protein pairs.

After searching for different feature extraction and selection methods, we used the iLearnPlus [46] method for extracting features and further selecting useful and important features with ANOVA[48] method. With ANOVA, we were able to select 10 different feature sets from protein sequences and 9 different feature sets from sRNA sequences.

With the features taken from the feature selection step, we evaluated three machine-learning methods to select the one with the highest 10-fold CV predict performance in terms of accuracy. XGBoost was the most accurate classifier, with a 10-fold CV accuracy of 0.948 ±0.02 on our bacterial sRNA-protein interaction data, and an average accuracy of 0.885 ±0.03 on four commonly used RPI data sets. This is comparable with other programs for RPI prediction. For example, RPITER, a program that uses sequences and structures as input data, reported an average accuracy of 0.88 ±0.11 on the same four data sets.

5.1 Future Work

There are several avenues for future work that could build upon the current research and address some of its limitations, like considering other species other than only bacteria. One direction is to explore other feature selection methods, another is to generate the negative instances with actual sRNA sequences instead of randomly selected genomic sequences, and another is to explore other machine learning approaches.

5.2 Project Code

ProNA code is an open-source model for predicting protein-sRNA interactions in bacteria, and its code is freely accessible at its GitHub repository, which can be found at `https://github.com/BioinformaticsLabAtMUN/ProNA`.

Bibliography

[1] Tanmay Dutta and Shubhangi Srivastava. Small RNA-mediated regulation in bacteria: a growing palette of diverse mechanisms. *Gene*, 656:60–72, 2018.

[2] Hal A Lewis, Kiran Musunuru, Kirk B Jensen, Carme Edo, Hua Chen, Robert B Darnell, and Stephen K Burley. Sequence-specific RNA binding by a nova KH domain: implications for paraneoplastic disease and the fragile X syndrome. *Cell*, 100(3):323–332, 2000.

[3] Shuping Cheng, Lu Zhang, Jianjun Tan, Weikang Gong, Chunhua Li, and Xiaoyi Zhang. DM-RPIs: Predicting ncRNA-protein interactions using stacked ensembling strategy. *Computational biology and chemistry*, 83:107088, 2019.

[4] Jingjing Wang, Yanpeng Zhao, Weikang Gong, Yang Liu, Mei Wang, Xiaoqian Huang, and Jianjun Tan. EDLMFC: an ensemble deep learning framework with multi-scale features combination for ncRNA–

protein interaction prediction. *BMC bioinformatics*, 22:1–19, 2021.

[5] Zhong-Hao Ren, Chang-Qing Yu, Li-Ping Li, Zhu-Hong You, Yong-Jian Guan, Yue-Chao Li, and Jie Pan. SAWRPI: A stacking ensemble framework with adaptive weight for predicting ncRNA-protein interactions using sequence information. *Frontiers in Genetics*, 13, 2022.

[6] Usha K Muppirala, Vasant G Honavar, and Drena Dobbs. Predicting RNA-protein interactions using only sequence information. *BMC bioinformatics*, 12(1):1–11, 2011.

[7] Ying Wang, Xiaowei Chen, Zhi-Ping Liu, Qiang Huang, Yong Wang, Derong Xu, Xiang-Sun Zhang, Runsheng Chen, and Luonan Chen. De novo prediction of RNA–protein interactions from sequence information. *Molecular BioSystems*, 9(1):133–142, 2013.

[8] V Suresh, Liang Liu, Donald Adjeroh, and Xiaobo Zhou. RPI-Pred: predicting ncRNA-protein interaction using sequence and structural information. *Nucleic acids research*, 43(3):1370–1379, 2015.

[9] Hai-Cheng Yi, Zhu-Hong You, De-Shuang Huang, Xiao Li, Tong-Hai Jiang, and Li-Ping Li. A deep learning framework for robust and accurate prediction of ncRNA-protein interactions using evolutionary information. *Molecular Therapy-Nucleic Acids*, 11:337–344, 2018.

[10] Wen-Jun Shen, Wenjuan Cui, Danze Chen, Jieming Zhang, and Jianzhen Xu. RPIRLS: quantitative predictions of RNA interacting with any protein of known sequence. *Molecules*, 23(3):540, 2018.

[11] Cheng Peng, Siyu Han, Hui Zhang, and Ying Li. RPITER: a hierarchical deep learning framework for ncRNA–protein interaction prediction. *International journal of molecular sciences*, 20(5):1070, 2019.

[12] Hai-Cheng Yi, Zhu-Hong You, Mei-Neng Wang, Zhen-Hao Guo, Yan-Bin Wang, and Ji-Ren Zhou. RPI-SE: a stacking ensemble learning framework for ncRNA-protein interactions prediction using sequence information. *BMC bioinformatics*, 21(1):1–10, 2020.

[13] Viplove Arora and Guido Sanguinetti. De novo prediction of RNA-protein interactions with graph neural networks. *RNA*, 28(11):1469–1480, 2022.

[14] Andrew Santiago-Frangos and Sarah A Woodson. Hfq chaperone brings speed dating to bacterial sRNA. *Wiley Interdisciplinary Reviews: RNA*, 9(4):e1475, 2018.

[15] Prajna R Kulkarni, Xiaohui Cui, Joshua W Williams, Ann M Stevens, and Rahul V Kulkarni. Prediction of CsrA regulating small RNAs in bacteria and their experimental verification in *Vibrio fischeri*. *Nucleic*

acids research, 34(11):3361–3369, 2006.

[16] Alexander J Westermann, Elisa Venturini, Mikael E Sellin, Konrad U Förstner, Wolf-Dietrich Hardt, and Jörg Vogel. The major RNA-binding protein ProQ impacts virulence gene expression in *Salmonella enterica serovar Typhimurium*. *MBio*, 10(1):e02504–18, 2019.

[17] Mikolaj Olejniczak and Gisela Storz. ProQ/FinO-domain proteins: another ubiquitous family of RNA matchmakers? *Molecular microbiology*, 104(6):905–915, 2017.

[18] C.W.J. Smith. *RNA-protein Interactions: A Practical Approach*. Practical approach series. Oxford University Press, 1998.

[19] Benjamin A Lewis, Rasna R Walia, Michael Terribilini, Jeff Ferguson, Charles Zheng, Vasant Honavar, and Drena Dobbs. PRIDB: a protein–RNA interface database. *Nucleic acids research*, 39(suppl_1):D277–D282, 2010.

[20] Stephen K Burley, Helen M Berman, Gerard J Kleywegt, John L Markley, Haruki Nakamura, and Sameer Velankar. Protein data bank (PDB): the single global macromolecular structure archive. *Protein crystallography: methods and protocols*, pages 627–641, 2017.

[21] Vera Pancaldi and Jürg Bähler. In silico characterization and

prediction of global protein-mRNA interactions in yeast. *Nucleic acids research*, 39(14):5826–5836, 2011.

[22] Tao Wu, Jie Wang, Changning Liu, Yong Zhang, Baochen Shi, Xiaopeng Zhu, Zhihua Zhang, Geir Skogerbø, Lan Chen, Hongchao Lu, et al. NPINTER: the noncoding RNAs and protein related biomacromolecules interaction database. *Nucleic acids research*, 34(suppl_1):D150–D152, 2006.

[23] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[24] Helen M Berman, Wilma K Olson, David L Beveridge, John Westbrook, Anke Gelbin, Tamas Demeny, Shu-Hsin Hsieh, AR Srinivasan, and Bohdan Schneider. The nucleic acid database. a comprehensive relational database of three-dimensional structures of nucleic acids. *Biophysical journal*, 63(3):751, 1992.

[25] V Suresh, K Ganesan, and S Parthasarathy. PDB-2-PB: a curated online protein block sequence database. *Journal of Applied Crystallography*, 45(1):127–129, 2012.

[26] Guohui Zheng, Xiang-Jun Lu, and Wilma K Olson. Web 3DNA—a web server for the analysis, reconstruction, and visualization of

three-dimensional nucleic-acid structures. *Nucleic acids research*, 37(suppl_2):W240–W246, 2009.

[27] Andreas R Gruber, Ronny Lorenz, Stephan H Bernhart, Richard Neuböck, and Ivo L Hofacker. The vienna RNA website. *Nucleic acids research*, 36(suppl_2):W70–W74, 2008.

[28] Hervé Abdi. Singular value decomposition (SVD) and generalized singular value decomposition. *Encyclopedia of measurement and statistics*, 907:912, 2007.

[29] C-W Chong, P Raveendran, and R Mukundan. The scale invariants of pseudo-zernike moments. *Pattern Analysis & Applications*, 6:176–184, 2003.

[30] Ryan Rifkin, Gene Yeo, Tomaso Poggio, et al. Regularized least-squares classification. *Nato Science Series Sub Series III Computer and Systems Sciences*, 190:131–154, 2003.

[31] Christophe Geourjon and Gilbert Deleage. SOPMA: significant improvements in protein secondary structure prediction by consensus prediction from multiple alignments. *Bioinformatics*, 11(6):681–684, 1995.

[32] Ying Huang, Beifang Niu, Ying Gao, Limin Fu, and Weizhong Li.

CD-HIT Suite: a web server for clustering and comparing biological sequences. *Bioinformatics*, 26(5):680–682, 2010.

[33] Hui Zhang, Huazhong Shu, Gouenou Coatrieux, Jie Zhu, QM Jonathan Wu, Yue Zhang, Hongqing Zhu, and Limin Luo. Affine legendre moment invariants for image watermarking robust to geometric distortions. *IEEE Transactions on Image Processing*, 20(8):2189–2199, 2011.

[34] Gianluca Corrado, Toma Tebaldi, Fabrizio Costa, Paolo Frasconi, and Andrea Passerini. RNAcommender: genome-wide recommendation of RNA–protein interactions. *Bioinformatics*, 32(23):3627–3634, 2016.

[35] Dong-Sheng Cao, Qing-Song Xu, and Yi-Zeng Liang. propy: a tool to generate various modes of chou's PseAAC. *Bioinformatics*, 29(7):960–962, 2013.

[36] Dong-Sheng Cao, Yi-Zeng Liang, Jun Yan, Gui-Shan Tan, Qing-Song Xu, and Shao Liu. PYDPI: freely available python package for chemoinformatics, bioinformatics, and chemogenomics studies. 2013.

[37] Bastiaan A van den Berg, Marcel JT Reinders, Johannes A Roubos, and Dick de Ridder. SPICE: a web-based tool for sequence-based protein classification and exploration. *BMC bioinformatics*, 15(1):1–10, 2014.

[38] Wei Chen, Xitong Zhang, Jordan Brooker, Hao Lin, Liqing Zhang, and Kuo-Chen Chou. PSEKNC-General: a cross-platform package for generating various modes of pseudo nucleotide compositions. *Bioinformatics*, 31(1):119–120, 2015.

[39] Dan Ofer and Michal Linial. PROFET: Feature engineering captures high-level protein functions. *Bioinformatics*, 31(21):3429–3436, 2015.

[40] Bin Liu, Fule Liu, Longyun Fang, Xiaolong Wang, and Kuo-Chen Chou. repDNA: a python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. *Bioinformatics*, 31(8):1307–1309, 2015.

[41] Jiawei Wang, Bingjiao Yang, Jerico Revote, Andre Leier, Tatiana T Marquez-Lago, Geoffrey Webb, Jiangning Song, Kuo-Chen Chou, and Trevor Lithgow. POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on pssm profiles. *Bioinformatics*, 33(17):2756–2758, 2017.

[42] Zhen Chen, Pei Zhao, Fuyi Li, André Leier, Tatiana T Marquez-Lago, Yanan Wang, Geoffrey I Webb, A Ian Smith, Roger J Daly, Kuo-Chen Chou, et al. iFeature: a python package and web server for features extraction and selection from protein and peptide sequences.

Bioinformatics, 34(14):2499–2502, 2018.

[43] Jie Dong, Zhi-Jiang Yao, Lin Zhang, Feijun Luo, Qinlu Lin, Ai-Ping Lu, Alex F Chen, and Dong-Sheng Cao. PYBIOMed: a python library for various molecular representations of chemicals, proteins and DNAs and their interactions. *Journal of cheminformatics*, 10(1):1–11, 2018.

[44] Bin Liu. BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Briefings in bioinformatics*, 20(4):1280–1294, 2019.

[45] Rafsanjani Muhammod, Sajid Ahmed, Dewan Md Farid, Swakkhar Shatabda, Alok Sharma, and Abdollah Dehzangi. PyFeat: a python-based effective feature generation tool for DNA, RNA and protein sequences. *Bioinformatics*, 35(19):3831–3833, 2019.

[46] Zhen Chen, Pei Zhao, Chen Li, Fuyi Li, Dongxu Xiang, Yong-Zi Chen, Tatsuya Akutsu, Roger J Daly, Geoffrey I Webb, Quanzhi Zhao, et al. iLearnPlus: a comprehensive and automated machine-learning platform for nucleic acid and protein sequence analysis, prediction and visualization. *Nucleic acids research*, 49(10):e60–e60, 2021.

[47] Robson P Bonidia, Douglas S Domingues, Danilo S Sanches, and André CPLF de Carvalho. MathFeature: feature extraction package for DNA, RNA and protein sequences based on mathematical

descriptors. *Briefings in bioinformatics*, 23(1):bbab434, 2022.

[48] Jason Brownlee. How to perform feature selection with numerical input data. *Machine Learning Mastery*, 2020.

[49] Sahar Melamed, Asaf Peer, Raya Faigenbaum-Romm, Yair E Gatt, Niv Reiss, Amir Bar, Yael Altuvia, Liron Argaman, and Hanah Margalit. Global mapping of small RNA-target interactions in bacteria. *Molecular cell*, 63(5):884–897, 2016.

[50] Ivana Bilusic, Niko Popitsch, Philipp Rescheneder, Renée Schroeder, and Meghan Lybecker. Revisiting the coding potential of the *E. coli* genome through Hfq co-immunoprecipitation. *RNA biology*, 11(5):641–654, 2014.

[51] Alexandra Sittka, Sacha Lucchini, Kai Papenfort, Cynthia M Sharma, Katarzyna Rolle, Tim T Binnewies, Jay CD Hinton, and Jörg Vogel. Deep sequencing analysis of small noncoding RNA and mRNA targets of the global post-transcriptional regulator, Hfq. *PLoS genetics*, 4(8):e1000163, 2008.

[52] Jai J Tree, Sander Granneman, Sean P McAteer, David Tollervey, and David L Gally. Identification of bacteriophage-encoded anti-sRNAs in pathogenic *Escherichia coli*. *Molecular cell*, 55(2):199–213, 2014.

[53] Yanjie Chao, Kai Papenfort, Richard Reinhardt, Cynthia M Sharma,

and Jörg Vogel. An atlas of Hfq-bound transcripts reveals 3 UTRs as a genomic reservoir of regulatory small RNAs. *The EMBO journal*, 31(20):4005–4019, 2012.

[54] Aixia Zhang, Daniel J Schu, Brian C Tjaden, Gisela Storz, and Susan Gottesman. Mutations in interaction surfaces differentially impact *E. coli* Hfq association with small RNAs and their mRNA targets. *Journal of molecular biology*, 425(19):3678–3697, 2013.

[55] Omar Torres-Quesada, Jan Reinkensmeier, Jan-Philip Schlüter, Marta Robledo, Alexandra Peregrina, Robert Giegerich, Nicolás Toro, Anke Becker, and Jose I Jiménez-Zurdo. Genome-wide profiling of Hfq-binding RNAs uncovers extensive post-transcriptional rewiring of major stress response and symbiotic regulons in *Sinorhizobium meliloti*. *RNA biology*, 11(5):563–579, 2014.

[56] Alexandre Smirnov, Konrad U Förstner, Erik Holmqvist, Andreas Otto, Regina Günster, Dörte Becher, Richard Reinhardt, and Jörg Vogel. Grad-seq guides the discovery of ProQ as a major small RNA-binding protein. *Proceedings of the National Academy of Sciences*, 113(41):11591–11596, 2016.

[57] Youssef El Mouali, Milan Gerovac, Raminta Mineikaitė, and Jörg Vogel. In vivo targets of *Salmonella* FinO include a FinP-like small

RNA controlling copy number of a cohabitating plasmid. *Nucleic Acids Research*, 49(9):5319–5335, 2021.

[58] Sahar Melamed, Philip P Adams, Aixia Zhang, Hongen Zhang, and Gisela Storz. RNA-RNA interactomes of ProQ and Hfq reveal overlapping and competing roles. *Molecular cell*, 77(2):411–425, 2020.

[59] Erik Holmqvist, Patrick R Wright, Lei Li, Thorsten Bischler, Lars Barquist, Richard Reinhardt, Rolf Backofen, and Jörg Vogel. Global RNA recognition patterns of post-transcriptional regulators Hfq and CsrA revealed by UV crosslinking in vivo. *The EMBO journal*, 35(9):991–1011, 2016.

[60] Yanping Han, Dong Chen, Yanfeng Yan, Xiaofang Gao, Zizhong Liu, Yaqiang Xue, Yi Zhang, and Ruifu Yang. Hfq globally binds and destabilizes sRNAs and mRNAs in *Yersinia pestis*. *Msystems*, 4(4):e00245–19, 2019.

[61] Philip Möller, Aaron Overlöper, Konrad U Förstner, Tuan-Nan Wen, Cynthia M Sharma, Erh-Min Lai, and Franz Narberhaus. Profound impact of Hfq on nutrient acquisition, metabolism and motility in the plant pathogen *Agrobacterium tumefaciens*. *PLoS One*, 9(10):e110427, 2014.

[62] Youssef El Mouali, Falk Ponath, Vinzent Scharrer, Nicolas Wenner,

Jay CD Hinton, and Jörg Vogel. Scanning mutagenesis of RNA-binding protein ProQ reveals a quality control role for the lon protease. *Rna*, 27(12):1512–1527, 2021.

[63] Michael Dambach, Irnov Irnov, and Wade C Winkler. Association of RNAs with *Bacillus subtilis* Hfq. *PloS one*, 8(2):e55156, 2013.

[64] Saskia Bauriedl, Milan Gerovac, Nadja Heidrich, Thorsten Bischler, Lars Barquist, Jörg Vogel, and Christoph Schoen. The minimal meningococcal ProQ protein has an intrinsic capacity for structure-based global RNA recognition. *Nature communications*, 11(1):2823, 2020.

[65] Bashir Saadeh, Clayton C Caswell, Yanjie Chao, Philippe Berta, Alice Rebecca Wattam, R Martin Roop, and David O'Callaghan. Transcriptome-wide identification of Hfq-associated RNAs in *Brucella suis* by deep sequencing. *Journal of bacteriology*, 198(3):427–435, 2016.

[66] Bork A Berghoff, Jens Glaeser, Cynthia M Sharma, Monica Zobawa, Friedrich Lottspeich, Jörg Vogel, and Gabriele Klug. Contribution of Hfq to photooxidative stress resistance and global regulation in *Rhodobacter sphaeroides*. *Molecular microbiology*, 80(6):1479–1495, 2011.

[67] Quan Zeng and George W Sundin. Genome-wide identification of Hfq-regulated small RNAs in the fire blight pathogen *Erwinia amylovora* discovered small RNAs with virulence regulatory function. *BMC genomics*, 15(1):1–19, 2014.

[68] Manuela Fuchs, Vanessa Lamm-Schmidt, Johannes Sulzer, Falk Ponath, Laura Jenniches, Joseph A Kirk, Robert P Fagan, Lars Barquist, Jörg Vogel, and Franziska Faber. An RNA-centric global view of *Clostridioides difficile* reveals broad activity of Hfq in a clinically important gram-positive bacterium. *Proceedings of the National Academy of Sciences*, 118(25):e2103579118, 2021.

[69] Thermo Fisher Scientific. `https://www.thermofisher.com/`. Accessed: 2023-03-14.

[70] Kotaro Chihara, Thorsten Bischler, Lars Barquist, Vivian A Monzon, Naohiro Noda, Jörg Vogel, and Satoshi Tsuneda. Conditional Hfq association with small noncoding RNAs in *Pseudomonas aeruginosa* revealed through comparative UV cross-linking immunoprecipitation followed by high-throughput sequencing. *MSystems*, 4(6):e00590–19, 2019.

[71] getfasta. `https://bedtools.readthedocs.io/en/latest/content/tools/getfasta.html`. Accessed: 2023-03-04.

[72] UniProt. https://www.uniprot.org/. Accessed: 2023-03-04.

[73] shuffle. https://bedtools.readthedocs.io/en/latest/content/tools/shuffle.html. Accessed: 2023-03-04.

[74] PseEIIP: Pseudo Electron-Ion Interaction Pseudopotentials of Trinucleotide (PseEIIP). https://www.rdocumentation.org/packages/ftrCOOL/versions/2.0.0/topics/PseEIIP. Accessed: 2023-03-08.

[75] Jia-Nan Sun, Hua-Yi Yang, Jing Yao, Hui Ding, Shu-Guang Han, Cheng-Yan Wu, and Hua Tang. Prediction of cyclin protein using two-step feature selection technique. *IEEE Access*, 8:109535–109542, 2020.

[76] ASDC: Adaptive skip dipeptide composition (ASDC). https://rdrr.io/cran/ftrCOOL/man/ASDC.html. Accessed: 2023-03-08.

[77] Emily L Gulliver, Brandon M Sy, Julia L Wong, Deanna S Deveson Lucas, David R Powell, Marina Harper, Jai J Tree, and John D Boyce. The role and targets of the RNA-binding protein ProQ in the gram-negative bacterial pathogen *Pasteurella multocida*. *Journal of Bacteriology*, 204(4):e00592–21, 2022.

[78] M Megroz. *The role of Hfq and sRNAs in regulation of Pasteurella*

multocida gene expression. PhD thesis, Monash University Clayton, Australia, 2020.