12-2022

ncRNA-protein Interaction Prediction using Language-based Features

Krishna Shah
University of New Orleans, kshah2@uno.edu

ncRNA-protein Interaction Prediction using Language-based Features

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
Computer Science

by

Krishna Shah

B.Sc. University of New Orleans

December, 2022

Abstract

Noncoding RNAs (ncRNAs) play a significant role in several fundamental biological processes by binding to RNA-binding proteins (RBPs); hence, it is necessary to study ncRNA-protein interaction (RPI). Several classic and deep-learning machine learning models have been proposed to predict RPI. These models first need to collect features of RNA and protein, such as physicochemical properties, secondary and tertiary structure, et cetera, before feeding them into the model. More recently, after the advancement of high throughput sequencing and the improvement in Natural Language Processing (NLP), transformer models like BERT-RBP and Evolutionary Scaling Model (ESM) can be trained to automatically extract feature representations, containing both low and high-level information, from RNA and protein sequences directly. This method could make manual feature collection optional. Hence, in this study, we compare the performance of such language-based features against manually created features to predict the interaction probability between a protein and an RNA.

Keywords: Machine Learning, Transformer, RNA-Protein Interaction, NLP, noncoding RNA

Acknowledgments

First and foremost, my deepest gratitude to my thesis advisor, Dr. Md Tamjidul Hoque for his continuous support and encouragement throughout my master's degree. Without his guidance, I would have neither started nor completed my thesis.

I would also like to thank Dr. Christopher Summa and Dr. Atriya Sen for taking the time to review my thesis and judge my defense.

Thank you Duaa Alawad and Md. Wasi Ul Kabir, for your invaluable advice and help during my research.

Finally, thank you to all my friends and families who have been a part of this journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Deoxy-ribonucelic acid (DNA), ribonucleic acid (RNA), and protein are the fundamental biomolecules in all living things. When they are in perfect harmony, they provide structure and function to organisms made of single cells to the ones made of millions of cells. On the other hand, when there is a disruption in the check and balance of these biomolecules, several maladies like cancer, obesity, genetic disorders, Alzheimer's disease, and so on can affect the organism. Many of the functions of these biomolecules depend on the synergy between them. For instance, transcription factors (TFs) bind to DNA to control the rate of transcription of a gene from DNA to RNA. Hence, several biological and computational methods have been developed to investigate the identity and the effects of the interaction between these molecules. After the proliferation of sequence data from the advancement in high throughput sequencing technologies, computational methods have been coming forward as a crucial contributor in this field.

One focus of such computational methods is to predict if an RNA and a protein molecule interact with each other. These methods differ from each other on two aspects: **a) Input Features**: These methods use different combinations of features like primary sequence conjoint triads, secondary structure, physicochemical properties, tertiary structures, *et cetera* as input. These features are curated from experimental data, created manually, or collected from another predictive model. **b) Training Model**: These methods use several machine learning models, ranging from classical and deep-learning methods.

More recently, two transformer models: ESM-1b and BERT-RBP can be trained on a huge sequence data of proteins and polynucleotides to learn their underlying features. These models treat the sequences as a language to encode the hidden features into a representation matrix. This representation could be used as an input feature in an RPI prediction model. In this thesis, we test the performance of automatic features from ESM-1b and BERT-RBP models to predict the interaction between noncoding RNA and protein. We test this with the performance of improved CTF from primary sequence and secondary structures. For the training model, we used a state-of-the-art model in RPI prediction. Using representation learning for feature generation provides similar performance in the task of RPI prediction. The use of such automatically generated features can simplify RPI prediction and other analyses of biosequences.

The organization of the thesis follows as mentioned here. Chapter 2 provides fundamental information about the biological significance of the interaction between RNA and protein and provides a literature review of biological and computational methods used in the RPI study. It also explains representation learning, Transformer, and machine learning models needed to understand the thesis thoroughly. Chapter 3 provides a comprehensive review of the datasets, details performance metrics, and explains how features are generated and models are designed. In chapter 4, we provide results from the study and discuss their implications. The thesis concludes with conclusive remarks on the use of language models in RPI prediction and provides insights into future works and improvements.

Chapter 2

Literature Review

This section provides information about previous works that have been done in the field of RNA-protein interaction prediction models. While doing so, it also provides the biological background for the significance of the interactions between biomolecules and develops the motivation for this study. The section reviews natural language models that are paramount in comprehending this thesis.

2.1 Significance of RNA Protein Interaction

The biological essence of living beings is founded upon a very delicate balance between several biomolecules, including nucleic acids and proteins. According to the central dogma of molecular biology, DNA, also considered the blueprint of life [12], stores the genetic code; DNAs are transcribed to messenger RNA (mRNA), and subsequently, mRNAs are translated to proteins [9] (Figure 2.1). Proteins are the eventual expressions of the genetic codes, and they provide structure and function to different cells and living organisms as a whole [26]. Akin to the Chinese philosophy of yin and yang, these biomolecules affect each other's life cycle through both positive and negative feedback mechanisms [38]. As an example of DNA-Protein interaction, transcription factors (TFs), a group of proteins, bind to DNA at their regulatory regions like promoters, enhancers, and inhibitors, thereby regulating the rate of transcription. Proteins can also bind to mRNAs and determine the mRNAs' fate through splicing, localization, transport and/or degradation [16, 30]. Similarly, proteins can bind to other proteins to modulate each other's activity.

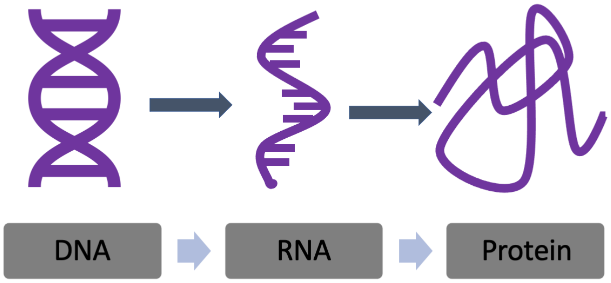


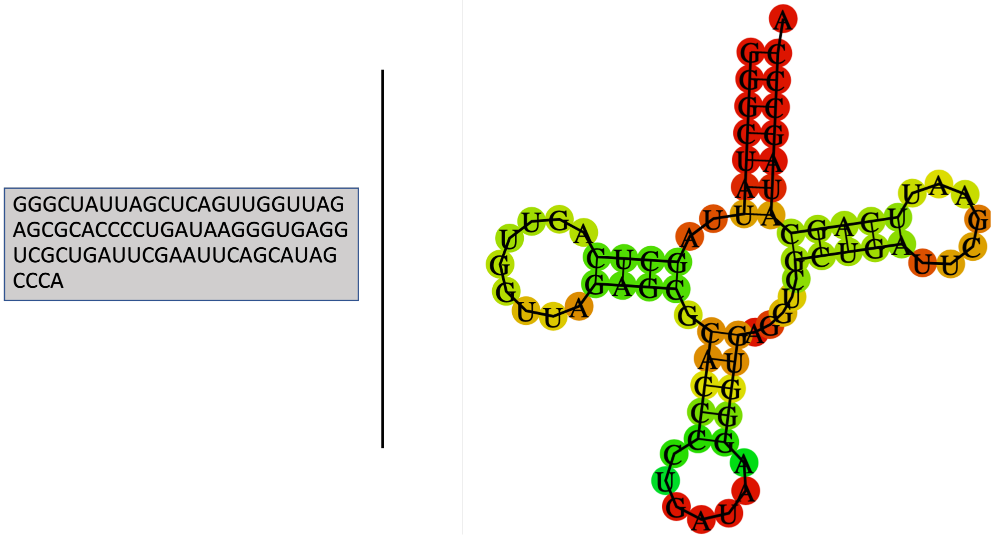
Figure 2.1: **Central Dogma of Molecular Biology**. *From left to right:* DNA stores the genetic code. DNA is transcribed to mRNA. mRNA translates to proteins.

In this thesis, our focus is on ncRNA-Protein interaction. There are two types of RNA: coding (mRNA) and noncoding (ncRNA). mRNAs are coding RNAs that get translated to proteins. They constitute only about 1-2 % of the total RNA content in mammals [7, 23, 15, 17]. As mentioned before, proteins bind to these mRNAs and determine the fate of the mRNAs' translation through splicing, localization, transport and/or degradation [16, 30]. The remaining large proportion of RNAs called noncoding RNAs (ncRNAs), bind with RNA-binding proteins (RBPs) to form a complex and regulate the expression of other DNAs and RNAs. They are important in several biological processes like embryonic development [3, 19], immune response [1], and cell cycle regulation [21], and are also implicated in several types of cancers [34]. Owing to their very large proportion and their contribution to a vast majority of regulatory mechanisms in living cells, it is very important that the interaction between these ncRNA and protein pairs is studied.

2.2 Proteins and RNAs

2.2.1 RNA

RNAs, a polynucleotide, is a chain of 4 nucleotides (adenine, uracil, guanine, and cytosine) in different combinations. Adenine has a propensity to hydrogen bond with uracil, and guanine has a propensity to hydrogen bond with guanine. As a result, a chain of RNA sequences can fold onto itself to form a secondary structure. Figure 2.2 represents the primary sequence of an RNA molecule and its predicted secondary structure. These secondary structures form naturally, and it can be seen how the protruding structures on the outside, rather than the nucleotides in the inner region, can be more available to bind to other molecules.

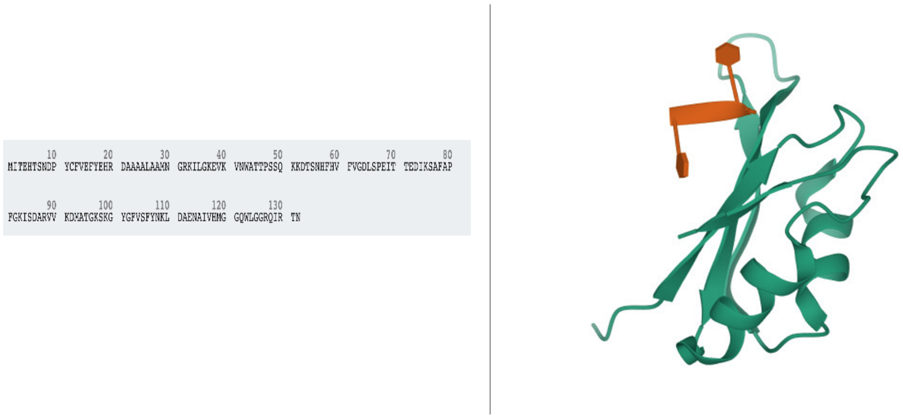


GGGCUAUUAGCUCAGUUGGUUAG
AGCGCACCCCUGAUAAGGGUGAGG
UCGCUGAUUCGAAUUCAGCAUAG
CCCA

Figure 2.2: **Ribonucleic Acids (RNA)** *Left:* Primary sequence of RNA. *Right:* Secondary structure of RNA. The strength of base pairs is represented on a blue-red scale.

2.2.2 Proteins

Protein, a polypeptide, is a chain of 20 different types of amino acids in different combinations. Although there are 20 amino acids, they can be classified into simple groups like polar, positively charged, negatively charged, hydrophobic, aromatic, and others. Because of hydrogen bondings, disulfide bridges, van der Waal's forces, hydrophobic interaction and ionic bonding, a polypeptide can fold onto itself to create a secondary structure and then its signature tertiary structure. Like RNA, some amino acid residues are hidden inside while some are exposed outside to interact with the environment. Figure 2.3 shows the primary structure of the TIAL1 protein and a part of its tertiary structure. We can also see the protein (green) interacting with an RNA molecule (orange).



```
        10         20         30         40         50         60         70         80
MITEHTSNDP YCFVEFYEHR DAAAALAAMN GRKILGKEVK VNWATTPSSQ KKDTSNHFHV FVGDLSPEIT TEDIKSAFAP

        90        100        110        120        130
FGKISDARVV KDMATGKSKG YGFVSFYNKL DAENAIVHMG GQWLGGRQIR TN
```

Figure 2.3: **Proteins** *Left:* Primary sequence of TIAL1 protein.*Right:* Tertiary structure of the protein sequence. The protein is shown to interact with nucleotides (orange) from an RNA molecule.

2.3 Current methods of RPI prediction

RNA-protein interaction pairs can be found experimentally using biochemical methods or can be predicted using computational models.

2.3.1 Non-computational methods

Several experimental methods have been developed to study the physical interaction between proteins and RNA in their native forms. These methods generally try to extract proteins bound to the RNA molecules (RNA-centric methods) or RNAs bound to a protein/ protein complex (protein-centric methods). Furthermore, these methods can be carried out in vivo, where the protein-RNA complexes in cells are cross-linked and further studied, or they can be carried out in vitro, where the proteins and RNAs of interest are allowed to bind together in a test tube or an array. Searching RPIs using such experimental techniques is a very expensive and tedious process and requires a lot of technical expertise on behalf of the researcher [27, 20, 37]. Although the biological methods of finding RNA-protein interaction pairs are long and tedious, different databases, including The Protein Data Bank (PDB), Protein-RNA Interface Database (PRIDB) and NPInter have been built upon the results from these experiments. Benchmark datasets curated from these databases are used to train and test the performance of computational methods.

2.3.2 Computational methods

To mitigate the drawbacks of experimental methods, several computational methods for RPI prediction have been proposed. With the advancement in high-throughput sequencing, there has been a rapid proliferation of sequence data of the biomolecules (DNA/RNA/Protein) [10]. These sequence data hold a lot of information about the biomolecules themselves; however they are not easily analyzed. Several computational methods have been proposed to tackle this issue. This section provides a review of such methods.

Muppirala *et al.* proposed RPISeq, which uses conjoint triad feature (CTF) vectors, previously used successfully in protein-protein interaction prediction [29], from protein and RNA sequences and feeds them to a random forest (RF) or support vector machine (SVM) classifier [24]. Belluci *et al.* put forward catRAPID, that uses physiochemical properties of protein and long noncoding RNAs (lncRNAs), including their secondary structure, hydrogen bonding and van der Waal's propensities [4, 2]. Wang *et al.* used similar features to Muppirala *et al.* but used Naive Bayes (NB) and Extended Naive Bayes (ENB) as classifiers [35]. Lu *et al.* proposed lncPro, which used similar input features as Belluci *et al.* but used a fisher linear discriminant approach for classification [22]. Suresh *et al.* forwarded RPI-Pred, which combines both primary sequence and tertiary structure features in the input vector and uses

SVM for classification [31]. Pan *et al.* proposed a deep learning model called IPMiner, which uses a stacked auto-encoder to extract primary sequence features from 3-mers RNA and 4-mers protein; this input vector is fed into an RF classifier and optimized by logistic regression (LR)-based ensemble model [25]. Similarly, Dai *et al.* proposed a novel method called complex features generated by non-linear transformation (CFRP) to extract features from the k-mers of RNA and protein and used RF to reduce feature dimension and perform prediction [11]. Wang *et al.* used a deep convolutional neural network to extract features and fed them to an extreme learning machine (ELM) for classification. Similarly, Cheng *et al.* stacked SVM, RF, and CNN to extract features and classify them from the primary sequences of RNA and protein [8]. Peng *et al.* proposed RPITER, which uses a hierarchical deep learning framework that uses an improved CTF coding method from primary sequence and structure information. Recently Wang *et al.* proposed a deep learning network called EDLMFC, that uses CNN-BLSTM-FC model on input vectors of improved CTF features from primary sequence, secondary and tertiary structure [33]. The main source of features for all these methods has been the database of primary, secondary, and tertiary structures of proteins and/or RNAs or some other predictive models. Having the secondary and tertiary structure for all the biomolecules is especially challenging because the experiments to capture these properties are very tedious and expensive. Similarly, other models that predict secondary and tertiary structures are not foolproof. Hence, some other methods of feature generation could be helpful.

2.4 Natural Language vs. Sequence data

With the advancement in NLP, there has been an increasing interest in bio-sequences because of their similarities with natural language. Natural language consists of alphabets, words, and is governed by grammatical and semantic rules. Similarly, sequences can be considered as made of smaller components. For instance, an mRNA sequence is made of four nucleotides

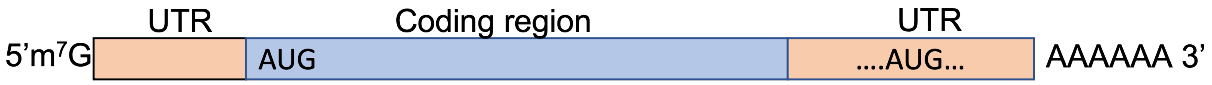
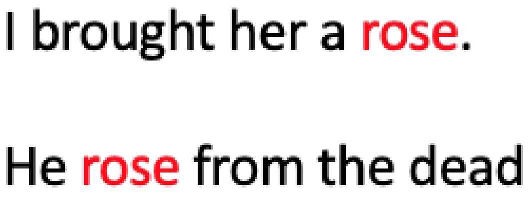


Figure 2.4: **Structure of mRNA:** An mRNA is read from 5' end to 3' end. Both on the left and right, there are untranslated regions (UTRs). The 3' end of mRNA contains a chain of As. Only the coding region in the middle gets translated to a protein. Translation starts when the 'AUG' code is read in the coding region.

(A, G, C and U); each mRNA sequence is just a stream of these four characters. We can also convert the sequence to conjoint k-mers (explained in detail in **Methods**) and treat them as words. These sequences can also be said to follow grammatical and semantic rules. We will take an example of an mRNA molecule (Figure 2.4). The mRNA in the figure is made of only As, Gs, Cs, and Us; however, there is a structure to it. The sequence always starts with an untranslated region (UTR) on the left, a coding region in the middle and a UTR on the right. Only the coding region gets translated to a protein sequence. Similarly, the pattern of these nucleotides conveys their meanings. 'AUG' at the beginning of the coding region always signals the end of translation; anywhere outside the coding region, it does not translate to any amino acid. Also, the poly-adenylated tail (chain of As in the 3' end) provides binding signals to transporting proteins that move the mRNA from the nucleus to the cytoplasm.



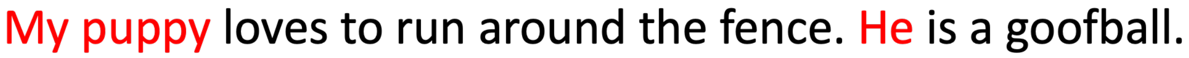
I brought her a rose.

He rose from the dead

Figure 2.5: **Context of words:** The meaning of the word 'rose' changes based on the context of words surrounding it.

Furthermore, the meanings of words in natural language depend on the context of the surrounding words. This gives rise to polysemy, where the same words can have different meanings depending on their context. Let's take an example of two sentences in Figure 2.5.

The word 'rose' in the first sentence means a literal flower whereas it means the act of getting up in the second sentence. Similarly, RNA codons (nucleotide triplets) can hold a different meaning in different places. While 'AUG' could mean nothing in UTR regions of mRNA, its presence in the coding region denotes the start of protein translation from the mRNA (Figure 2.4.



My puppy loves to run around the fence. He is a goofball.

Figure 2.6: **Distal relationship:** Although "my puppy" and "he" are parts of different sentences, both point to the same animal.

Moreover, words in natural language influence other words that are very far from themselves. For instance, *he* in Figure 2.6 refers to the puppy, although it is far away from it. Similarly, we can see nucleotides at extreme ends interacting with each other in the secondary structure of an RNA (Figure 2.2), and thus influencing each other.

With these parallelisms between natural language and biosequences, it can be argued that they have at least a few features in common, if not most. Hence, there has been an interest in using language models in sequence analysis.

2.5 Representation Learning

Representation learning is a very important class of machine learning method that is used to learn the underlying characteristics of raw data and present the findings in an n-dimensional vector space. It allows the system to create representations that can be used for feature detection or any other classification tasks. We can take an example of colors to explain what the representation looks like (Figure 2.7). In a 3D space where each axis represents red, blue, and green, we can create a 3D vector to represent a color of any choice. Since color is an abstract concept, it is very advantageous to represent it as a feature vector. The fundamental properties of the colors are available in such feature vectors. Using representation learning

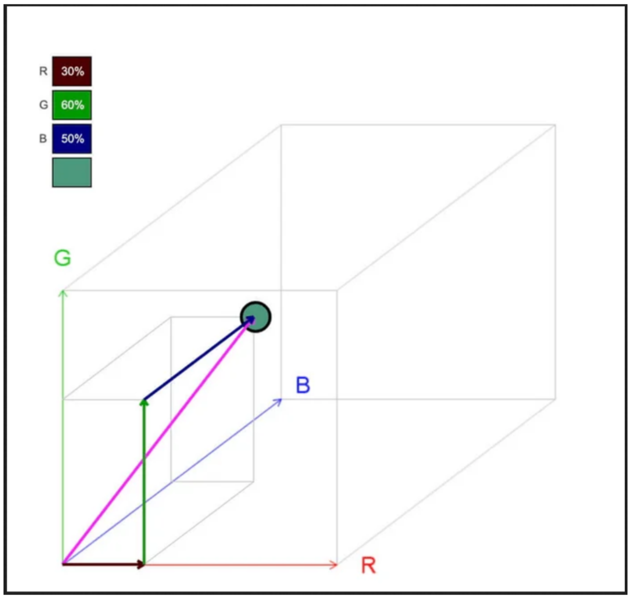


Figure 2.7: **Representing colors in a 3D space.** All colors can be represented as a vector in 3 dimensions.

methods, these vectors are created automatically without human intervention. Thus, it reduces the amount of domain-specific feature engineering. For instance, let's assume we are developing a machine learning model to identify different types of birds from their images. We may consult and ornithologist and create a model/method to extract a few features like feather color, beak length, leg size, wingspan, *et cetera*. Deciding on these features can be a grueling process; Even so, we cannot be sure that we have all the features. However, if we let a representation learning model learn features itself, it can look at each pixel and find the features that it calculates are important for the distinction between birds. This method simplifies the feature engineering process.

Similarly, representation learning can also be used to decompose natural language (or biosequence) into underlying features. Representation learning in NLP can be used to create a real-time vector, called a distributed vector, from the raw data like words or characters [5]. Such representation vectors preserve the semantic relationship between different words. In other words, terms that are like each other or have some relationships are represented closer together, and the ones that are different in meaning with no relationships are very far apart in the vector space. For example, in a 3D vector space created with representation learning for 10,000 words (Figure 2.8), words related to religion/societal structures will get grouped close together. Representation learning can also be used in sequence data to learn the underlying properties of the biosequences. We discuss some models that can produce a representation matrix from the sequence input.

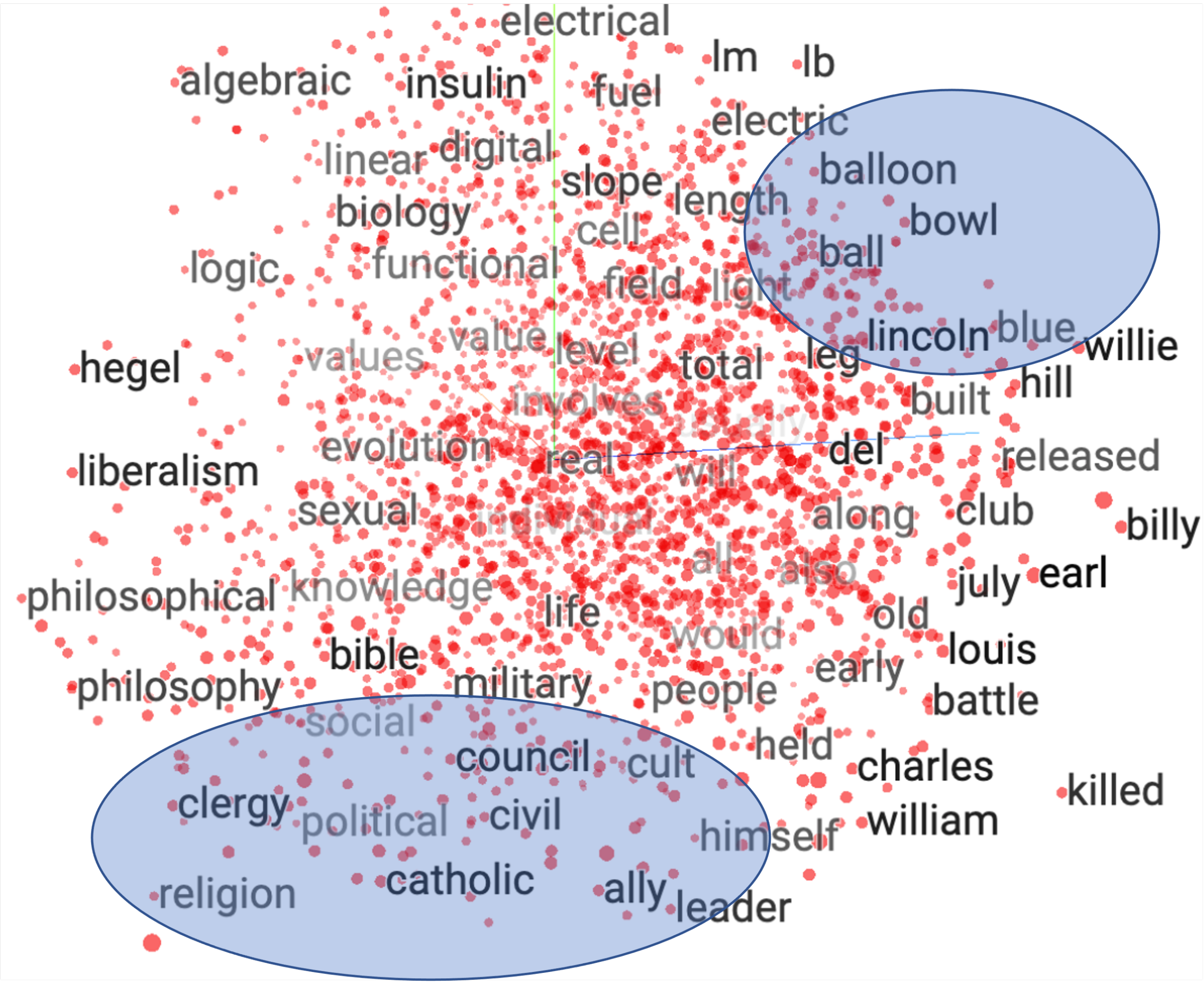


Figure 2.8: **Example of representation learning using word2vec**. Ten thousand words were represented using word2vec in a 3 dimension. Similar words related to religion and politics cluster together on the bottom left. Words related to some games are clustered on the top-right.

2.5.1 Transformers

The Transformer is a deep learning model used in NLP that uses encoder-decoder architecture with self-attention mechanism. Transformer architecture is shown in Figure 2.9. The encoder on the left is made of a stack of identical layers. Each layer consist of two sublayers: multi-head self-attention and a fully-connected feed-forward network. After each sublayer, there is a layer normalization to the sum of residual from the previous sublayer and the output from the current sublayer.

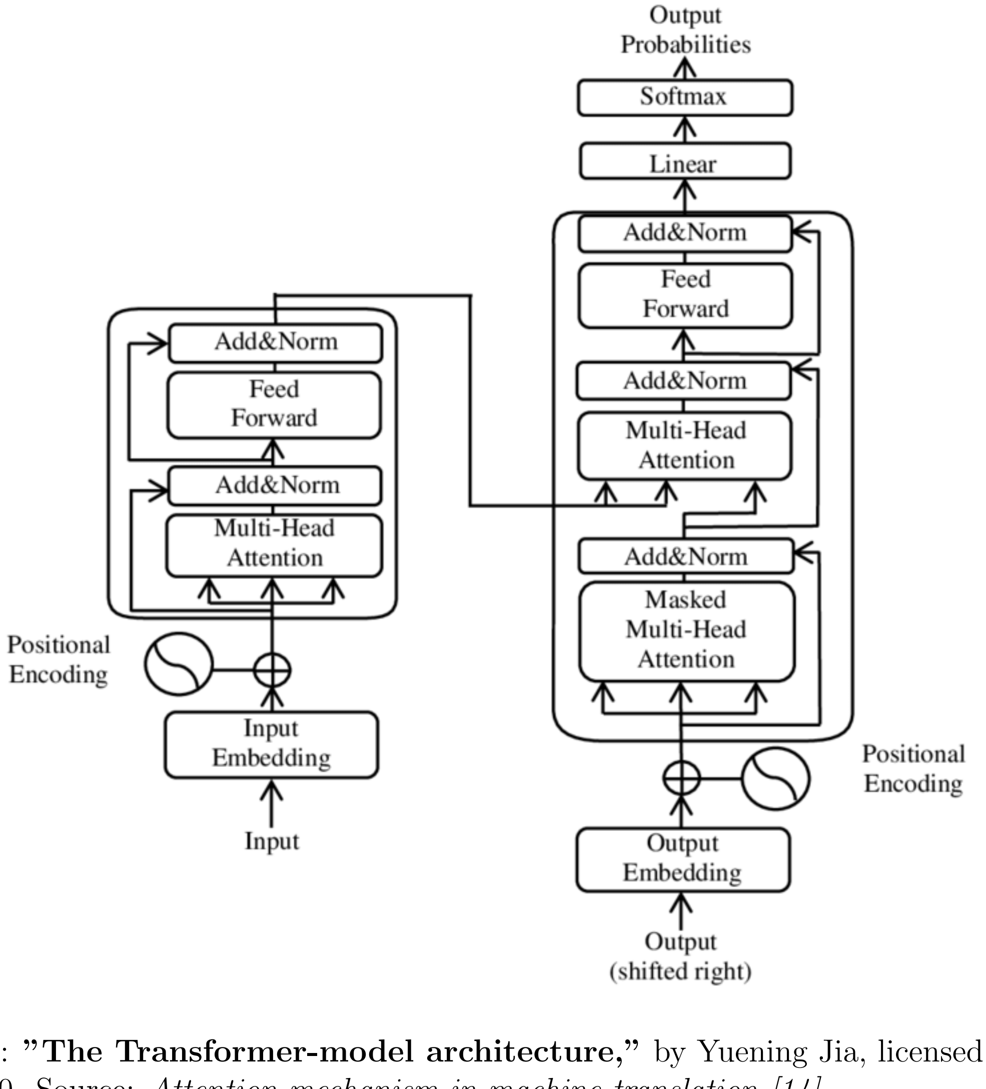


Figure 2.9: **"The Transformer-model architecture,"** by Yuening Jia, licensed under CC BY 3.0. Source: *Attention mechanism in machine translation [14].*

The decoder architecture is made up of a stack of identical layers like in the encoder.

Each layer consists of three sublayers, two similar to the encoder architecture; it also contains a masked multi-head attention layer. After each sublayer, there is a layer normalization to the sum of residual from the previous sublayer and the output from the current sublayer.

A Transformer can encode an input sequence into a representational vector matrix. First, each word in an input sequence is converted to its vector embeddings. Since we are dealing with the data sequence, each input point's position needs to be preserved. This is done by injecting positional encoding into the input embedding matrix. This then feeds into a multi-head self-attention layer where the relationship matrix is created for each position against the other matrix. This self-attention mechanism makes the Transformer model better than other sequential models like RNN (Recurrent Neural Network) and LSTM (Long Short Term Memory). Because of the self-attention mechanism, transformers can remember data points very far away, so they don't lose information. It is also very easy to traverse in both directions of the sequence. This is very important in learning features in language and sequence models, especially when the words very far away can affect the given word, like in RNA and protein sequences. The encoder layer by itself is enough to generate representations of sequences.

In the decoder layer, the output sequence is converted to a vector of output embedding, and positional encoding is injected into it. Then, the masked multi-head attention layer calculates attention among each position, but the representations are masked from every position except the ones we are trying to use for prediction; as a result, the decoder does not see what it is trying to predict. After this sublayer, it is very much like the encoder layer. This sublayer's output and the encoder's output feed into the multi-head self-attention layer. At the end of the decoder, a softmax function is used to make predictions [32].

2.5.2 DNABERT

DNABERT is a Bidirectional Encoder Representation from Transformers (BERT) based model that was trained on the human reference genome to extract the underlying semantics and context in the DNA sequence [13]. Its architecture is shown in Figure 2.10. The training dataset consisted of sequences that were 5 to 510 nucleotides long and were subsequently converted to k-mers tokens ranging from 3-mers to 6-mers. During the pre-training, DNABERT used masked language modeling by masking about 15 to 20 percent contiguous tokens and letting the model to predict the masked sequence through self-supervised learning based on the remaining available nucleotides. The pre-training model can be used to generate general representations of the nucleotide sequences.

In the next steps, the pre-trained model was fine-tuned to very small task-specific labeled data for prediction of promoters, splice sites, and transcription factor binding sites, and the model achieved state-of-the-art or comparative performance. Upon further analysis using DNABERT-viz, a tool to visualize the regions in input sequences that were contributing the model decision, the model representations showed high attentions weights to the promoter regions, splice cites and transcription factor binding sites in their respective tasks. Furthermore, the fine-tuned model was also shown to generalize in mouse model, underscoring the generalizability of the model.

2.5.3 BERT-RBP

Considering only one nucleotide difference between DNA and RNA (thymine vs. uracil, respectively), Yamada *et al.* modified the DNABERT model to predict if an RNA sequence is protein-binding or not [36]. Its architecture is shown in Figure 2.11. This model, called BERT-RBP, used the pre-trained model from DNABERT and fine-tuned them on positive and negative samples of RNA sequences involved in interaction with proteins. This representation from this fine-tuned model extracted information about RNA transcript type (5'UTR, 3'UTR, intron, and CDS) and RNA secondary structure through attention analysis. Thus, this model was a suitable candidate for extracting RNA features.

2.5.4 ESM-1b

The Evolutionary Scaling Model (ESM-1b) is also a BERT-based model that was selected from a series of BERT models called Evolutionary Scale Modeling (ESM). It was trained on a very large corpus of amino acid sequences, specifically 86 billion amino acids across 250 million polypeptide sequences spanning evolutionary diversity [28]. In the pre-training, each amino acid residue was treated as a word to create a model of 34-layers. The embedding from this 34-layer model was able to capture the secondary and tertiary structure of the sequence from just the primary sequence. It could also predict the remote homology and residue to residue contact from this representation. Thus, this model was a suitable candidate for extracting protein features.

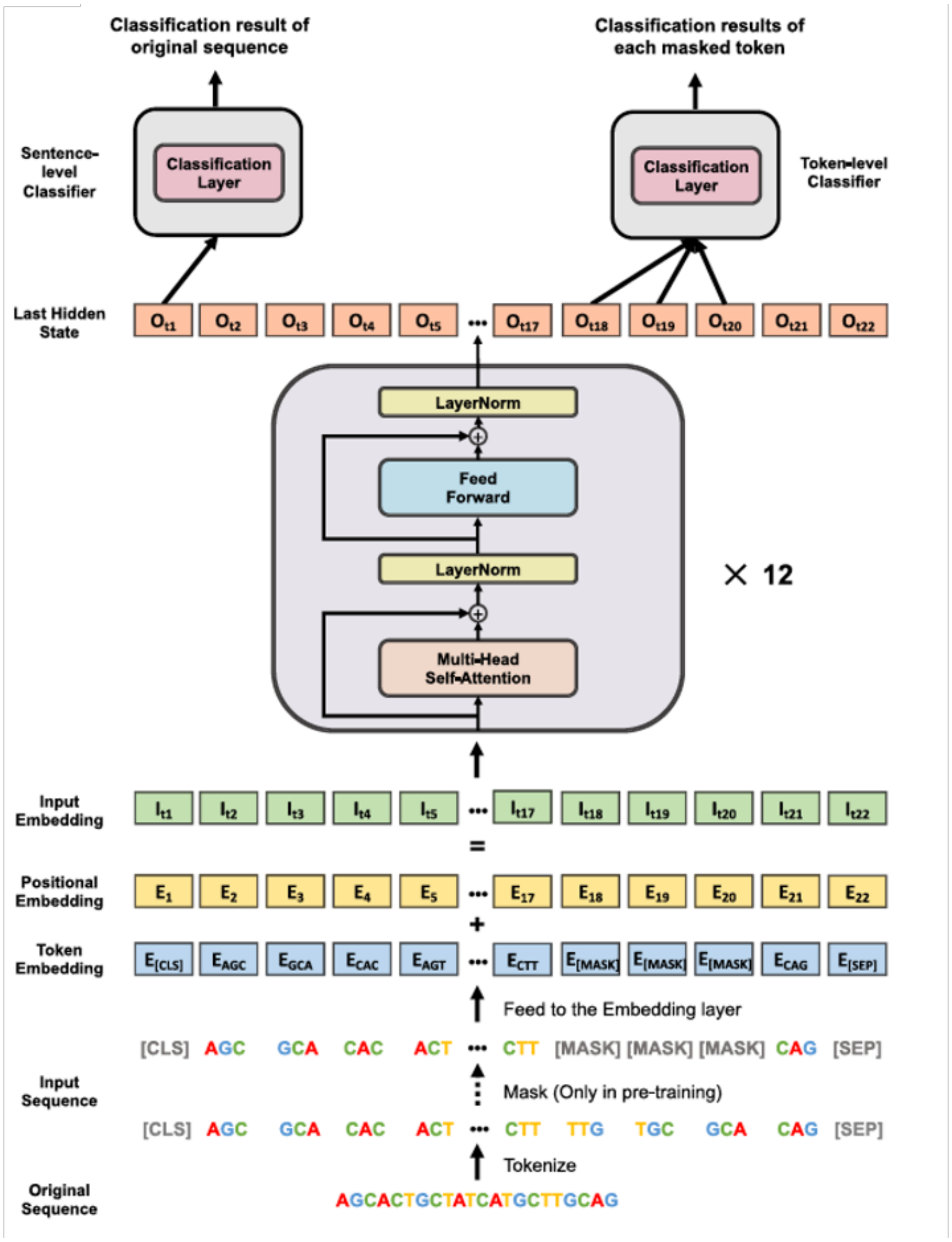

Classification result of original sequence
Classification results of each masked token
Sentence-level Classifier
Classification Layer
Classification Layer
Token-level Classifier
Last Hidden State
Ot1 Ot2 Ot3 Ot4 Ot5 ••• Ot17 Ot18 Ot19 Ot20 Ot21 Ot22
LayerNorm
Feed Forward
LayerNorm
Multi-Head Self-Attention
× 12
Input Embedding
It1 It2 It3 It4 It5 ••• It17 It18 It19 It20 It21 It22
=
Positional Embedding
E1 E2 E3 E4 E5 ••• E17 E18 E19 E20 E21 E22
+
Token Embedding
E[CLS] EAGC EGCA ECAC EAGT ••• ECTT E[MASK] E[MASK] E[MASK] ECAG E[SEP]
Feed to the Embedding layer
Input Sequence
[CLS] AGC GCA CAC ACT ••• CTT [MASK] [MASK] [MASK] CAG [SEP]
Mask (Only in pre-training)
[CLS] AGC GCA CAC ACT ••• CTT TTG TGC GCA CAG [SEP]
Tokenize
Original Sequence
AGCACTGCTATCATGCTTGCAG

Figure 2.10: **DNABERT Model Architecture** by Ji Yanrong *et al.* Source: *DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome [13].*

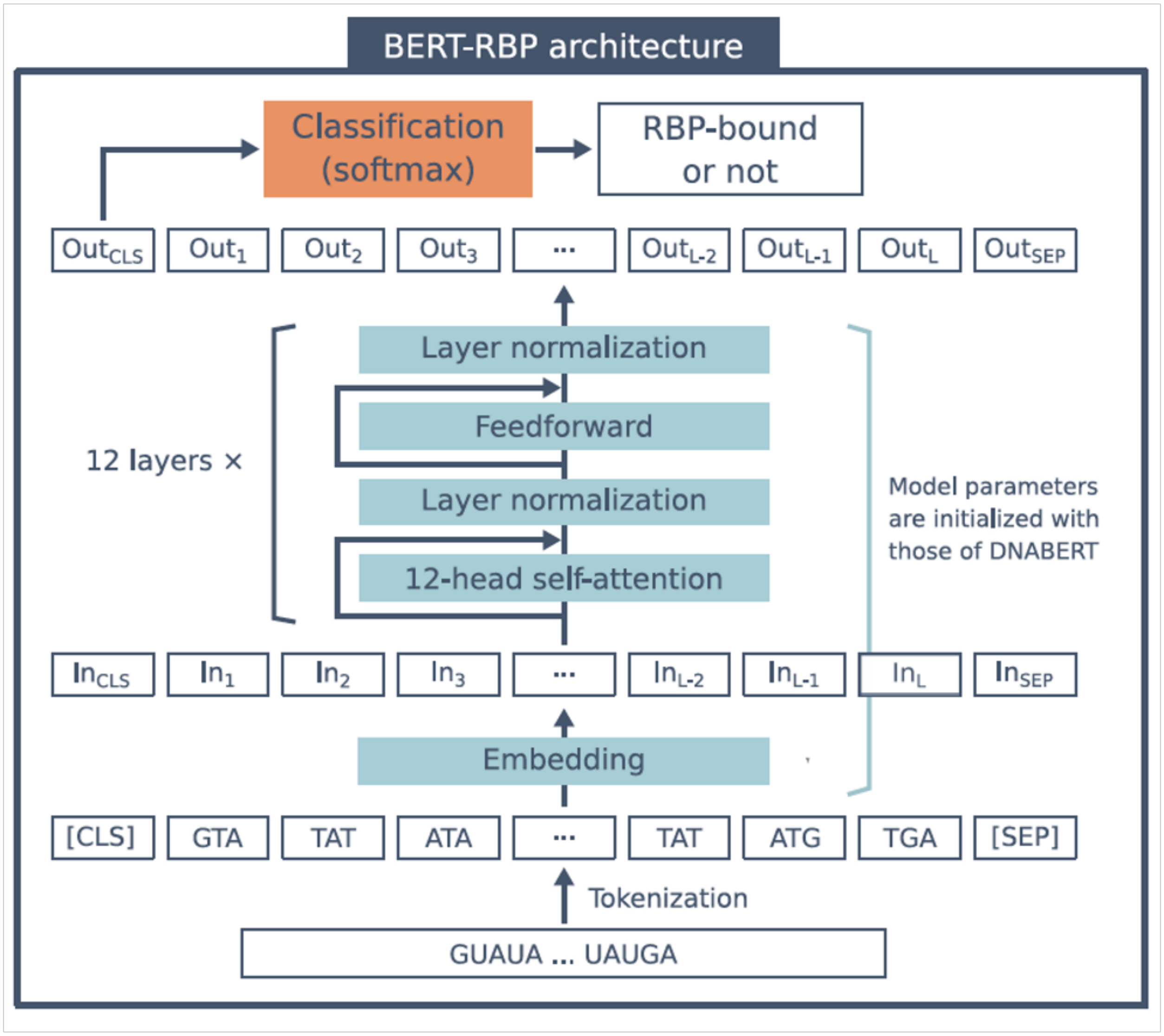


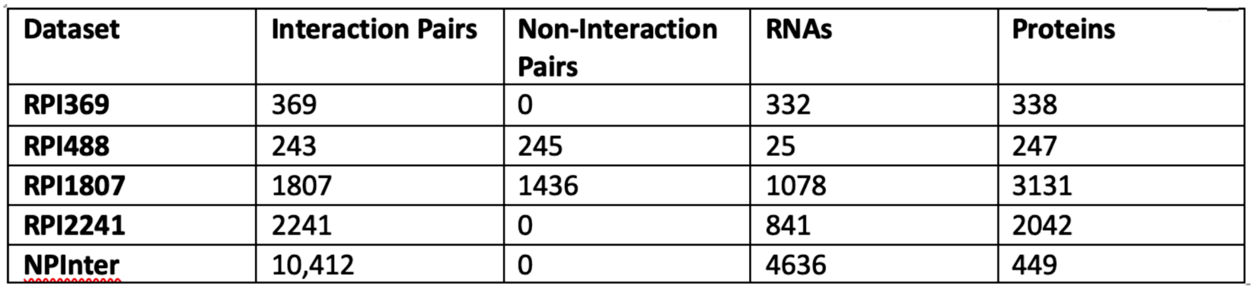
Figure 2.11: **BERT-RBP Model Architecture**, by Yamada *et al.*, licensed under CC BY 4.0. Source : *Prediction of RNA–protein interactions using a nucleotide language model [36].*

Chapter 3

Methodology

3.1 Benchmark Datasets

All datasets (Table 3.1) in this study were downloaded from `https://github.com/Pengeace/RPITER`. RPI369, RPI488, RPI1807, and RPI2241 were extracted from Protein-RNA Interface Database (PRIDB) [18] and the Protein Data Bank (PDB) [6]. The details about the datasets are provided here.



Dataset	Interaction Pairs	Non-Interaction Pairs	RNAs	Proteins
RPI369	369	0	332	338
RPI488	243	245	25	247
RPI1807	1807	1436	1078	3131
RPI2241	2241	0	841	2042
NPInter	10,412	0	4636	449

Table 3.1: **Benchmark Datasets**.

3.1.1 RPI2241

RPI2241 is a benchmark non-redundant dataset that consists of RNA-protein interacting pairs extracted from PRIDB. PRIDB is a protein-RNA interface database calculated from the protein-RNA complexes in PDB. Using an 8 Å distance cutoff, RNA-proteins interacting pairs were extracted from the 943 RNA-protein complexes in PRIDB. The original 943 RNA-protein complexes consisted of 9689 protein chains and 2074 RNA chains. RPI2241 consisted of 952 and 443 of these protein and RNA chains, respectively. In deriving these sequences, these rules were used. The proteins must be at least 25 residues long. The RNAs must be at least 15 nucleotides long. Two protein sequences were said redundant if they have >= 30% sequence identity. Similarly, two RNA sequences were said redundant if they have >= 30% sequence identity. They were discarded if the redundant protein chains interacted with similar RNA sequences. Following the same logic, if two RNA sequences were redundant and they interacted with similar protein sequences, they were discarded. As a result, the RPI2241 dataset with 2241 positive pairs was created.

A balanced dataset with 2241 non-interacting RNA-protein pairs or negative pairs was generated using these rules. First, RNA and proteins from 943 protein-RNA complexes in PRIDB were randomly paired. If both the RNA and protein contain at least 30% similarity with the positive pairs, they were discarded.

3.1.2 RPI369

About 40% of RNA-Protein complexes in PDB are made of ribosomal structures and since the PRIDB database is derived from PDB, it is similarly biased. Thus, a subset of RPI2241 which does not include these ribosomal proteins or ribosomal RNAs was created. This dataset is called RPI369. It contains 338 protein chains and 332 RNA chains. Negative pairs are generated as explained above.

3.1.3 RPI488

It is a benchmark non-redundant dataset that includes lncRNA-protein interaction pairs. lncRNAs (long noncoding RNAs) have sizes greater than 200 nucleotides. The dataset was created from 18 different lncRNA-Protein complexes in PDB. Using a 5.0 Å distance cutoff, lncRNA-proteins pairs were divided into positive and negative pairs. If the distance is smaller than the threshold of 5.0 Å, the pairs are categorized as positive. Otherwise, they are categorized as negative. This creates a redundant dataset. Redundant pairs were discarded using the following rules. Two protein sequences were said redundant if they have >= 90% sequence identity. Similarly, two RNA sequences were said redundant if they have >= 90% sequence identity. If redundant protein chains interacted with similar RNA sequences, they were discarded. Following the same logic, if two RNA sequences were redundant and they interacted with similar protein sequences, they were discarded. As a result, a dataset with 245 positive lncRNA-protein interaction pairs and 243 negative pairs was created. The dataset contains 25 different protein chains and 247 different RNA chains.

3.1.4 RPI1807

RPI1807 dataset is a benchmark non-redundant dataset created from the Nucleic acid Database (NDB) and Protein-RNA Interface Database (PRIDB). NDB contains RNA-protein complexes, and PRIDB contains RNA-Protein atomic interfaces. Protein chains longer than 25 residues and RNA chains longer than 15 nucleotides were selected. Using a 3.40 Å distance cutoff, RNA-proteins interacting pairs were divided into positive and negative pairs. If the distance is smaller than the threshold of 3.40 Å, the pairs are categorized as positive. Otherwise, they are categorized as negative. This creates a redundant dataset. Redundant pairs were discarded using the following rules. Two protein sequences were said redundant if they have >= 30% sequence identity. Similarly, two RNA sequences were said to be redundant if they have >= 30% sequence identity. If redundant protein chains interacted with similar RNA sequences, they were discarded. Following the same logic, if two RNA sequences were redundant and they interacted with similar protein sequences, they were discarded. As a result, 1807 positive RNA-protein interaction pairs and 1436 negative pairs were created. The dataset contains 1807 different protein chains and 3131 different RNA chains.

3.1.5 NPInter v2.0

The NPInter v2.0 dataset is a benchmark non-redundant dataset created from the NPInter database and new literature. These pairs have been experimentally verified rather than depending on their distances in the complex. It contains all noncoding RNA-Protein interactions except from rRNAs and tRNAs. The dataset is comprised of 10412 positive RNA-protein interaction pairs. The dataset contains 449 different protein chains and 4636 different RNA chains.

A balanced dataset with 10412 non-interacting RNA-protein pairs or negative pairs was generated using these rules. First, RNA and proteins in the positive samples were randomly paired. If the RNA shared >= 80% similarity and the protein shared >= 40% similarity with a positive ncRNA-Protein sample, this pair is discarded from the negative sample.

3.2 Model Architecture

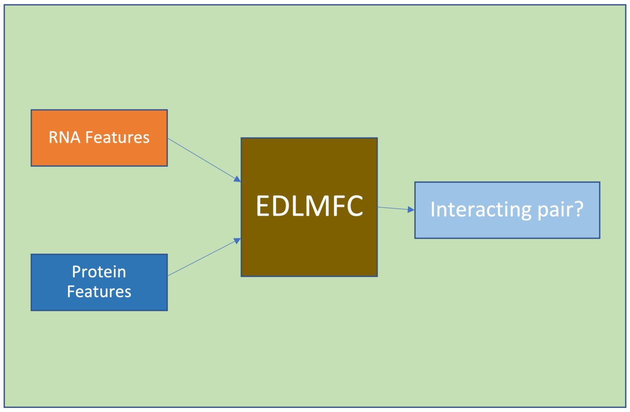


Figure 3.1: **Simple Model Architecture**. RNA features and protein features are fed into a machine learning model. The model predicts whether the protein-RNA pair bind to each other or not.

The overall architecture in all our experiments can be represented in Figure 3.1. First RNA features and protein features are extracted in different ways. These features are passed through a state-of-the-art EDLMFC model (Figure 3.2). First, each of the RNA and protein feature vectors are passed through a convolutional neural network and, subsequently, BLSTM to produce new RNA and protein feature vectors, respectively. These features are then concatenated and passed through a fully connected neural network. The final layer provides the classification output by passing fully connected layers output to a softmax function. A probability value of 0.5 or more represent positive interaction and a value of less than 0.5 represents non-interacting pairs.

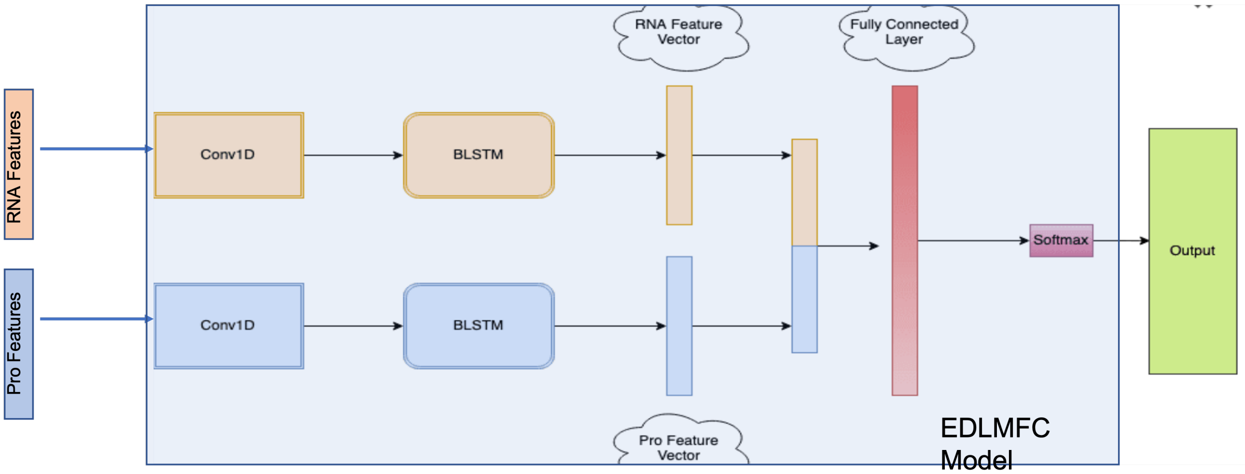


Figure 3.2: **Full Model Architecture**. A state-of-the-art EDLMFC model was used.

3.3 RPITER Feature Generation

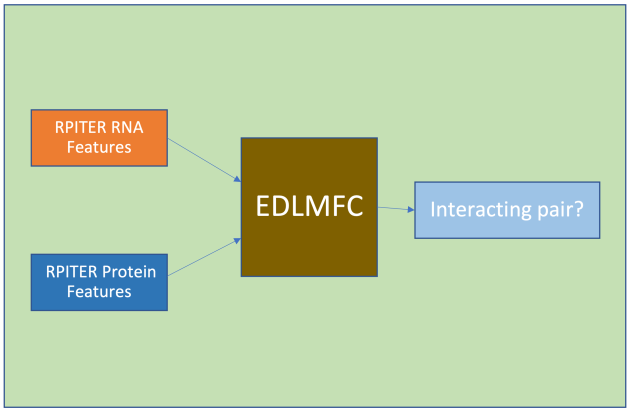


Figure 3.3: **Baseline**. RPITER features of RNA and protein were fed into the EDLMFC model.

For the baseline, we fed RPITER features into the EDLMFC model (Figure 3.3). RPITER uses improved conjoint triad features (CTF) from the primary sequence and secondary structure of both RNA and proteins.

3.3.1 CTF

Instead of treating each residue in a sequence as an input, they can also be represented as k-mers, as shown in Figure 3.4. A sliding window of size k is moved along the sequence to produce a k-mers at each position of the sequence. As a result, the sequence of size n produces n-k CTFs. The frequency of each possible CTFs is represented in a frequency vector; this acts as an input feature. Most popularly, RNA and protein sequences have been encoded into CTFs with k=4 and k=3, respectively.

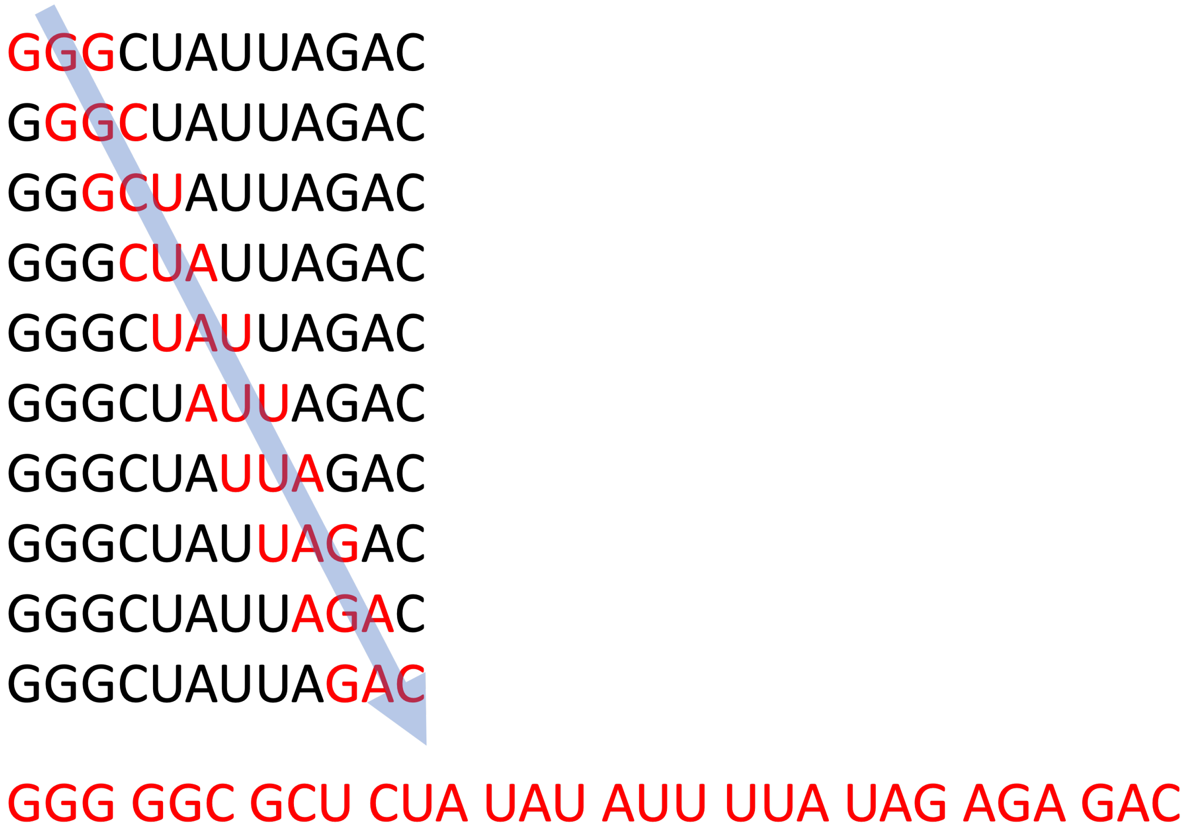


Figure 3.4: **Conjoint Triad Features from RNA sequence**. A sliding window of size 3 (red) moves along the sequence to produce the final CTF sequence.

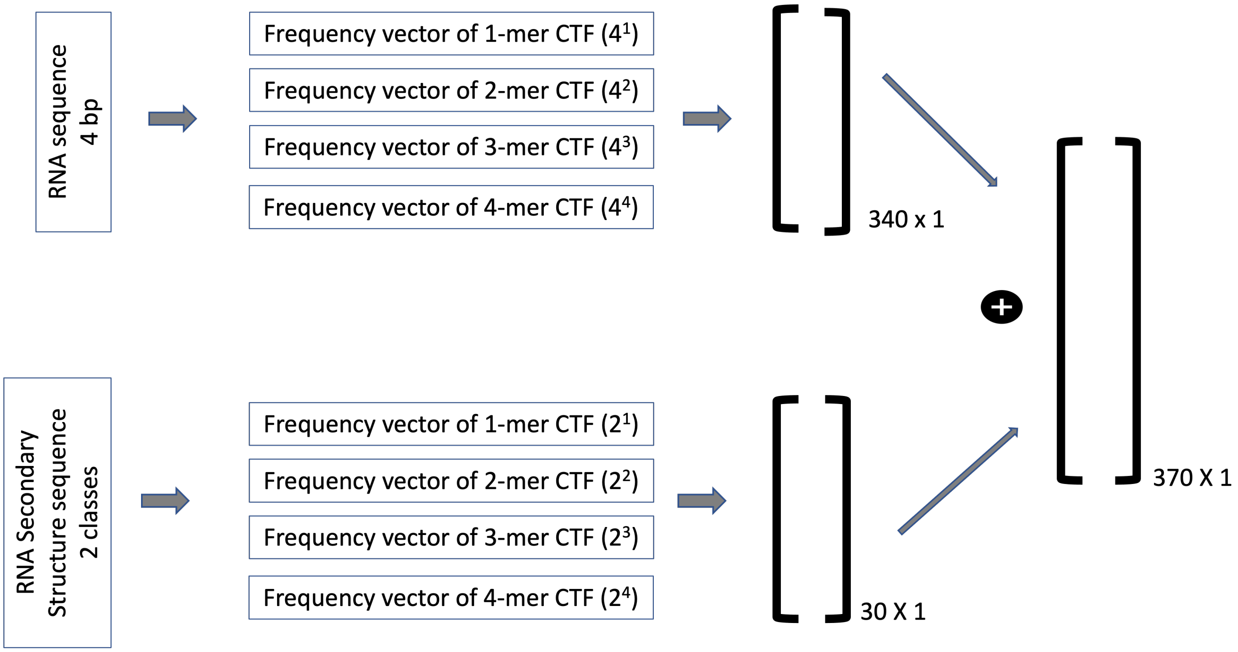
3.4 Feature Extraction

3.4.1 Improved CTF

Improved CTF adds to the concept of a regular CTF. Instead of creating a frequency vector for only k-mers CTFs, a frequency vector for 1-mers to k-mers CTFs is used as an input vector.

3.4.2 RNA Features

Improved 4-mers CTFs were created for both primary sequence (represented by 4 nucleotides) and secondary structure sequence (represented by 2 classes) to produce a final input vector of length 370 (Figure 3.5).



RNA sequence
4 bp
Frequency vector of 1-mer CTF (4¹)
Frequency vector of 2-mer CTF (4²)
Frequency vector of 3-mer CTF (4³)
Frequency vector of 4-mer CTF (4⁴)
340 x 1
370 X 1
RNA Secondary
Structure sequence
2 classes
Frequency vector of 1-mer CTF (2¹)
Frequency vector of 2-mer CTF (2²)
Frequency vector of 3-mer CTF (2³)
Frequency vector of 4-mer CTF (2⁴)
30 X 1

Figure 3.5: **Extraction of RNA features from RPITER**.

3.4.3 Protein Features

Improved 3-mers CTFs were created for both primary sequence (represented by 7 classes) and secondary structure sequence (represented by 3 classes) to produce a final input vector of length 438 (Figure 3.6).

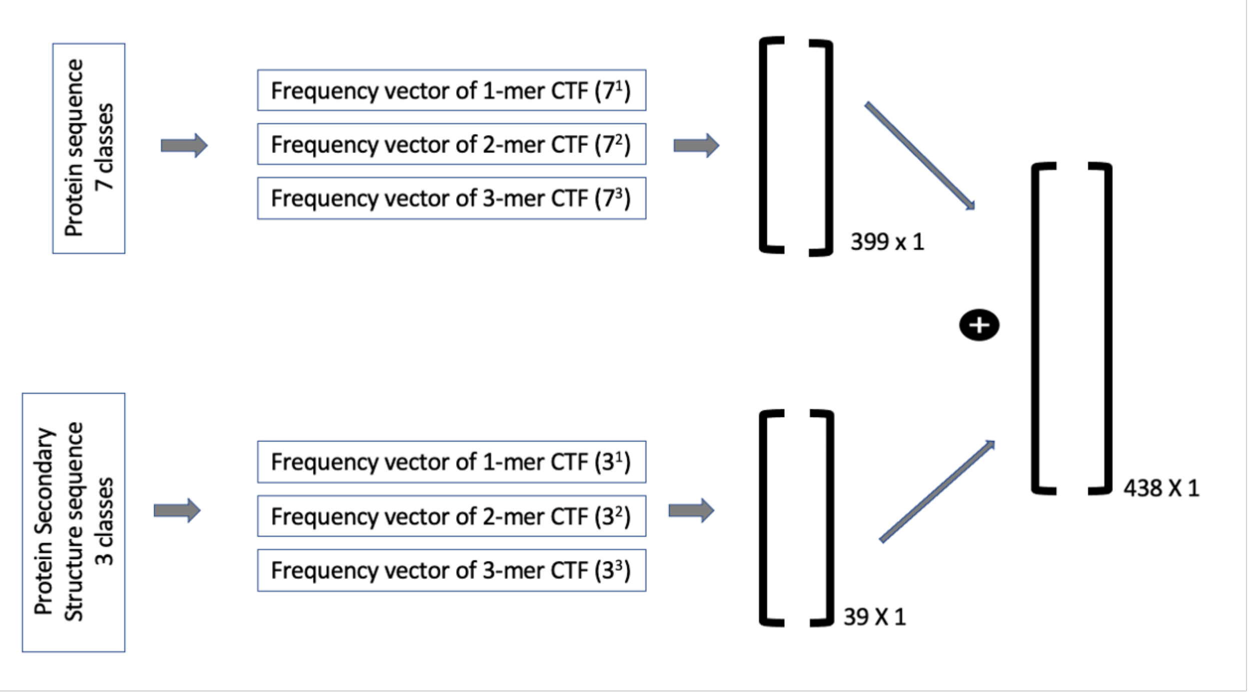

Protein sequence 7 classes
Frequency vector of 1-mer CTF (7^1)
Frequency vector of 2-mer CTF (7^2)
Frequency vector of 3-mer CTF (7^3)
399 x 1
438 X 1
Protein Secondary Structure sequence 3 classes
Frequency vector of 1-mer CTF (3^1)
Frequency vector of 2-mer CTF (3^2)
Frequency vector of 3-mer CTF (3^3)
39 X 1

Figure 3.6: **Extraction of Protein features from Improved CTF**.

3.5 Language-based Feature Generation

For the model with language-based features, we fed RNA features from BERT-RBP and protein features from ESM-1b models into the EDLMFC model (Figure 3.7). This feature set containing ESM-1b and BERT-RBP features will henceforth be referred as ESM/BERT.

3.5.1 BERT-RBP RNA Features

We downloaded 4 different pre-trained DNABERT models from `https://github.com/jerryji1993/DNABERT`. Each model had been pretrained on DNA CTF sequences with k values ranging

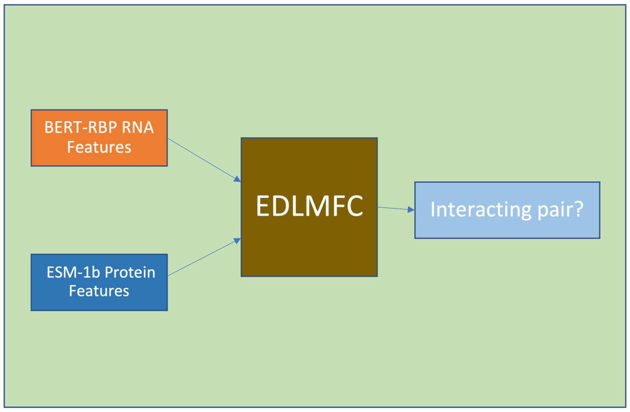


Figure 3.7: **Language-based model**. BERT-RBP RNA features and ESM-1b protein features were fed into the EDLMFC model.

from 3 to 6. These models were named DNABERT_k-mers for different values of k ranging from 3 to 6. Each value of k represents different CTF lengths in the input DNA sequence.

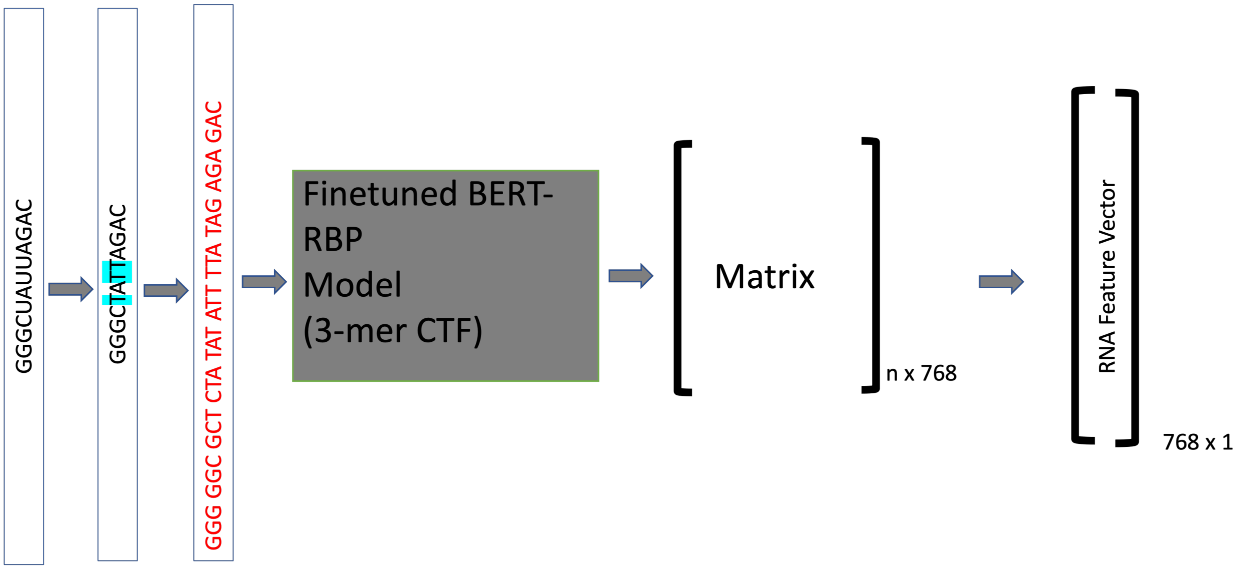

GGGCUAUUAGAC
GGGCTATTAGAC
GGG GGC GCT CTA TAT ATT TTA TAG AGA GAC
Finetuned BERT-RBP Model (3-mer CTF)
Matrix
n x 768
RNA Feature Vector
768 x 1

Figure 3.8: **Extraction of RNA features from BERT-RBP**. The diagram shows the extraction process for a model trained on 3-mers CTFs. So, the CTFs used here are also of size 3. The same process can be done for other CTFs.

We fine-tuned each pre-trained DNABERT model using RNA sequences represented with appropriate size CTFs. For instance, pretrained DNABERT_3mers model was fine-tuned with RNA sequences represented as 3-mers CTFs. All the U nucleotide residues in the sequence were converted to T to make the RNA sequence compatible with DNABERT model.

Fine-tuning was done on the task of classifying if a given RNA sequence binds to any protein or not. We used RNA sequences from a single protein-RNA complex of an mRNA-binding protein TIAL1. Only the ncRNA sequences that bind to TIAL1 were used to fine-tune the model, and fine-tuning was done for only 3 epochs because the pre-training process took a lot of time, up to 20 hours for each k-mers model. This fine-tuned model was used to produce RNA features from RNA sequences in training and testing for RPI prediction.

The process of extracting RNA-feature vector from fine-tuned BERT-RBP model is shown in Figure 3.8. First, U's in the RNA sequence are replaced with T's. Next the sequence is converted to CTFs. When the CTF sequence is fed to fine-tuned BERT-RBP model, a representation matrix in 768 dimensions is produced. We average the values at each position of the sequence to get a final RNA vector of size 768. In total, 4 input vectors were created from each RNA sequence for each k-mer values from 3 to 6.

3.6 ESM-1b Protein Features

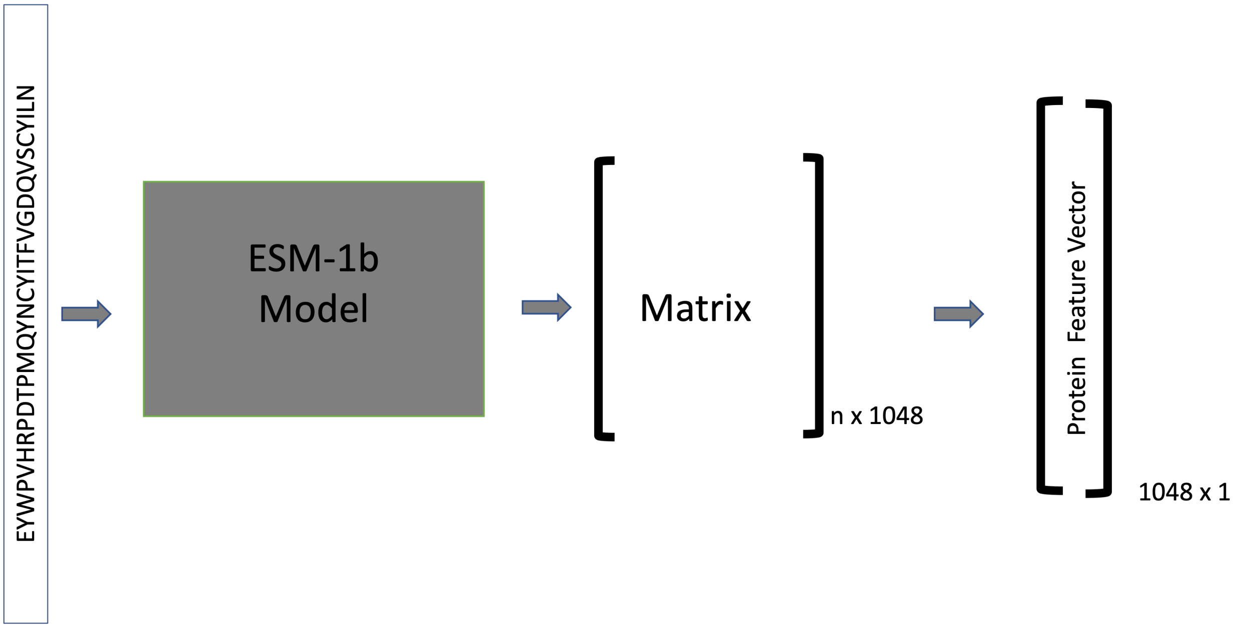


Figure 3.9: **Extraction of Protein features from ESM-1b**.

ESM-1b model was used directly to generate protein representation features (Figure 3.9).

First, a protein sequence, without transforming to CTFs, was passed through the ESM-1b model.

When a protein sequence is fed into this model, the model produces a representation matrix in 1048 dimensions. We average the values at each position of the sequence to get a final protein vector of size 1048. For protein sequences longer than 1022 residues, the sequence was broken down into chunks of size 1022 or less and stacked together. Then the values were averaged at each position to generate the final protein features vector.

3.7 Bonus: Combined Feature Generation

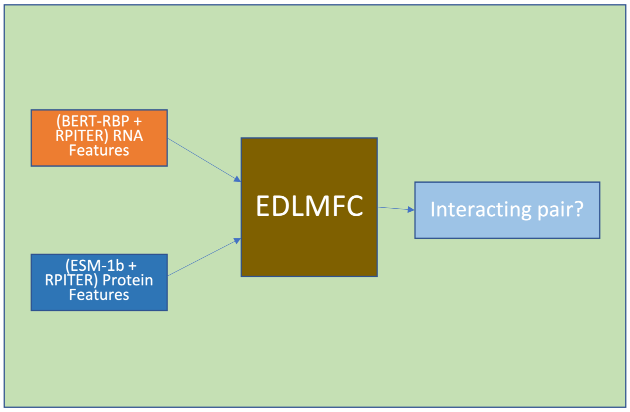

(BERT-RBP + RPITER) RNA Features
(ESM-1b + RPITER) Protein Features
EDLMFC
Interacting pair?

Figure 3.10: **Combined Features**.

As a bonus, we concatenated RNA features from RPITER and BERT-RBP to produce a combined RNA feature and we concatenated protein features from RPITER and ESM-1b. We fed these features to the EDLMFC model (Figure 3.10).

3.7.1 All models

In total, there were 9 total feature sets used (Figure 3.11). There is one baseline model with RPITER features (left panel), 4 ESM/BERT feature models (middle panel), 1 for each fine-tuned BERT-RBP model, and 4 combined feature models (right model). Each model was trained with each dataset.

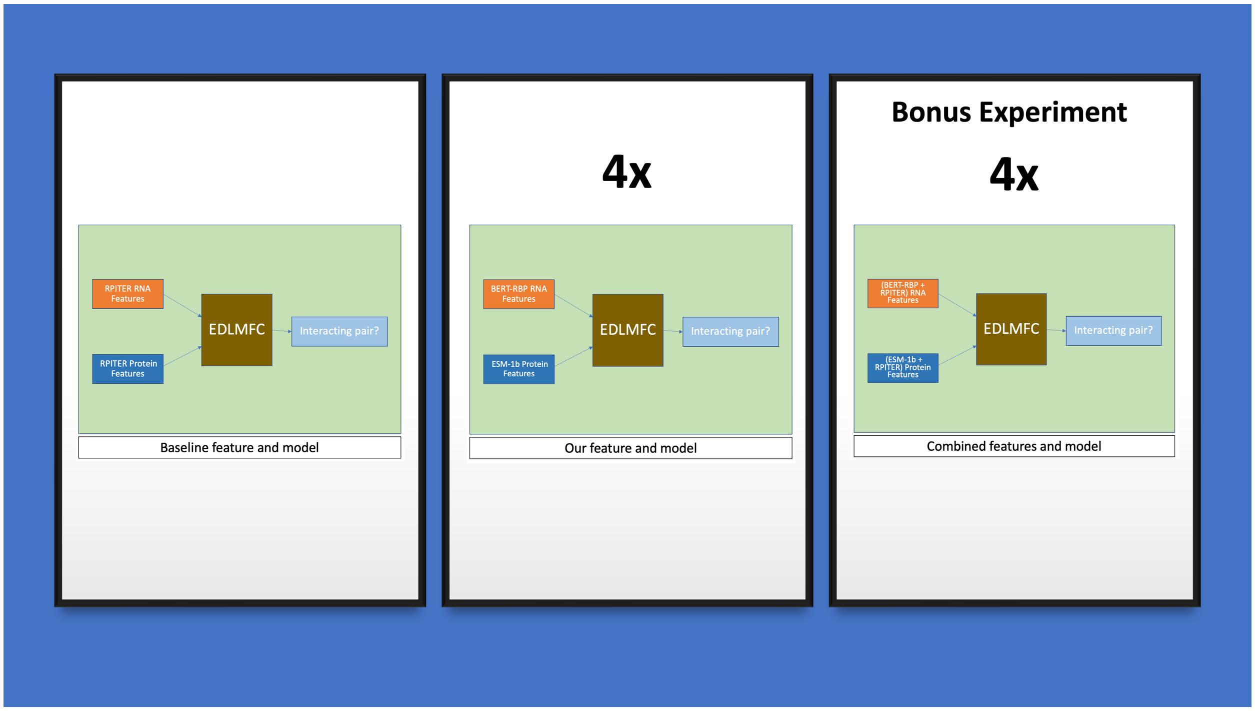


Figure 3.11: **Overview of all models used**. One baseline model with RPITER features was created (left). 4 models with protein features from ESM-1b and RNA features from BERT-RBP (using 4 different fine-tuned models pretrained on k-mers of 3 to 6) were used (middle). 4 more models where the features from RPITER and ESM/BERT were combined were used (right).

3.8 Performance Metrics

5-fold cross-validation (CV) was used to evaluate the performance of the model and to compare it with other results. In k-fold cross-validation, the dataset is randomly distributed among k different groups. Then, for each fold from 1 to k, the given fold is held out for testing while the remaining folds are used for training. After training and testing the model for k times using different fold datasets, the error estimates of the model among the iterations are averaged. The averaged error estimate represents the error rate of the model.

Based on the true nature of a protein-RNA pair and its classification, the output can be described as follows.

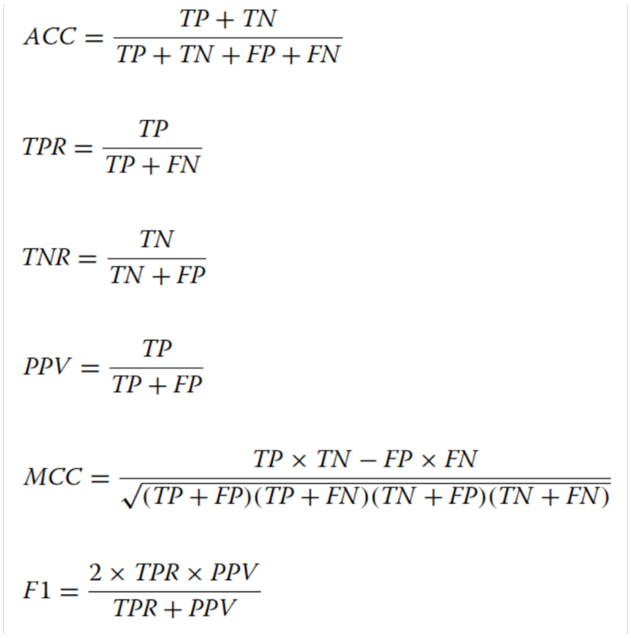
True Positive (TP): The pairs of RNA and Protein that interact with each other and are classified as interacting pairs.

True Negative (TN): The pairs of RNA and Protein that do not interact with each other and are classified as not-interacting pairs.

False Positive (FP): The pairs of RNA and Protein that do not interact with each other but are classified as interacting pairs.

False Negative (FN): The pairs of RNA and Protein that interact with each other but are classified as non-interacting pairs.

The performance metrics in Figure 3.12 were calculated using these four parameters.



$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

$$TPR = \frac{TP}{TP + FN}$$

$$TNR = \frac{TN}{TN + FP}$$

$$PPV = \frac{TP}{TP + FP}$$

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

$$F1 = \frac{2 \times TPR \times PPV}{TPR + PPV}$$

Figure 3.12: **Performance Metrics**.

Chapter 4

Results and Discussion

4.1 Results

We compared the performance of the model when the features were from RPITER (baseline), ESM-1b and BERT-RBP (ESM/BERT) for 4 k-mers values, and from RPITER and ESM/BERT combined (Combined) for 4 k-mers values. The comparisons were done for each of five datasets.

In RPI369 dataset, RPITER features performed the worst (accuracy of 0.52) than any other datasets or any other features (Figure 4.1). In comparison, generally both the ESM/BERT features and combined features showed improved accuracy. Among ESM/BERT features, 4-mers features performed the best with an increased accuracy of 7.8%. 6-mers ESM/BERT was an exception where these features had reduced accuracy by 7.8% as compared to the RPITER features. All combined features performed better than baseline and ESM/BERT features; the best performer (3-mers features) increased the accuracy by 30.6%.

In RPI488 dataset, the baseline RPITER features produced an accuracy of 0.68 (Figure 4.2). In comparison, the ESM/BERT features showed reduced accuracy (-8.1%, -1.9%, -0.3%, and -1.6% respectively). On the other hand, the combined features had more than 10% improved accuracy from 3-mers, 5-mers, and 6-mers features. 4-mers combined features showed a decline of 1.5%.

In RPI1807 dataset, the baseline RPITER features produced an accuracy of 0.97 (Figure 4.3). In comparison, the ESM/BERT features had similar accuracy, within +/- 0.1%. On the other hand, combined features showed a slight decrease in accuracy ranging from -0.1% to -0.8%.

In RPI2241 dataset, the baseline RPITER features produced an accuracy of 0.86 (Figure 4.4). In comparison, the ESM/BERT features showed an increased accuracy from 1.9% to 2.9%. Similarly, combined features consistently increased accuracy from 2.2% to 2.4%.

In NPInter dataset, the baseline RPITER features produced an accuracy of 0.95 (Figure 4.5). In comparison, the ESM/BERT features performed similarly, with improvements up to 0.3% and degradation up to -0.4%. Similarly, combined features showed accuracy improvement from -0.1% to 0.5

When we took the average of performance along all datasets, the baseline RPITER features produced an accuracy of 0.81 (Figure 4.6). In comparison, ESM/BERT features showed decline in accuray up to -1.0% and improvement up to 1.3%. On the other hand, combined features consistently increased accuracy from 3.6% to 6.5%.

In summary, compared to baseline RPITER features, both ESM/BERT and combined features had a very similar performance in all datasets except for some exceptions (Figure 4.1 - 4.5). In the RPI488 dataset, the combined features performed better than both RPITER and ESM/BERT features (Figure 4.2). In the RPI369 dataset, the overall performance was low; nevertheless, the ESM/BERT features performed better than the RPITER base features, and the combined features performed better than both the RPITER and ESM/BERT features (Figure 4.1). On average, the combined features performed slightly better than the other two feature sets (Figure 4.6).

4.2 Discussion

We chose to use the EDLMC model (CNN-BLSTM-FN) because it was a state-of-the-art model and performed better than the RPITER model. However, EDLMFC only used a subset of RNA-protein pairs in RPITER datasets for training and testing. These subsets only contained the RNA sequences for which the secondary structures could be predicted; furthermore, only the proteins associated with these RNAs were included. Since we wanted to test our models/features on a large dataset, we used RPITER datasets and their features for baseline. As a result, the performance of the baseline model with RPITER features was poor compared to EDLMFC's published metrics. This could mean that EDLMFC does not perform well only on the subset of datasets. Because we used the EDLMFC model for all our experiments and the only differences were input features, we can still compare the significance of the features used in our experiments.

For the most part, as compared to RPITER features, the features from BERT-RBP and ESM-1b models showed similar performances in all datasets. Only in RPI369 the ESM/BERT and combined features perform better than RPITER base features. First, the overall low performance with all feature sets in RPI369 maybe because this dataset was the smallest and the models could not learn fully from a small training dataset. Moreover, RPI369 is a subset of RPI2241 dataset; it does not include ribosomal proteins or ribosomal RNAs. The overall performance is very high in RPI2241 as compared to RPI369; although the large training size could have the performance, it could also mean that the models learn ribosomal proteins' and RNAs' features better. In RPI369, ESM/BERT features performed better than RPITER features, which could mean that ESM-1b and BERT-RBP models can extract more information from non-ribosomal proteins and RNAs. Since the combined features produced even better results, it could mean that RPITER features ESM/BERT features encoded somewhat different information in RPI369 and RPI 488.

Also, it cannot be guaranteed that the combined features perform better than the RPITER or ESM/BERT features separately. Only in RPI488 and RPI369, the two smallest datasets, the combined features produced noticably improved performance than the other features. This suggests that in smaller datasets, ESM-1b and BERT-RBP features do not learn all features that are present in RPITER features. But as the training datasets become larger, they perform at the same level as RPITER features, indicating that they have automatically learned features in RPITER feature sets.

There is no single winner for the choice of k-mers representation of RNAs in ESM/BERT model. So, any combination of k-mers for RNA and proteins used to train NLP models could extract similar information given a large dataset. This provides a solution to deciding which value of k in CTF provides the best feature; after all, the language-based models learn similarly with all k values.

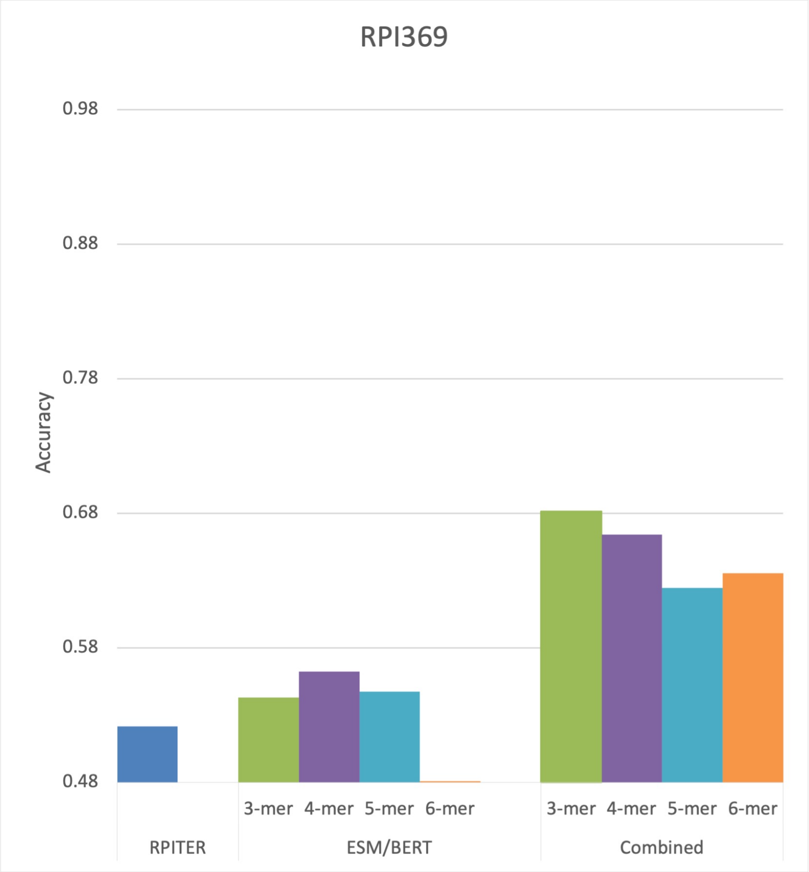
The biggest takeaway from these experiments can be that ESM/BERT features can capture secondary structure information (manually encoded in RPITER features) from the RNA and protein sequences. In most instances, BERT-RBP and ESM-1b features performed similar to the improved CTF features. Since improved CTF depends on both primary sequence and secondary structure, we can assume that the features from the two transformer models capture at least the primary sequence and secondary structure information. ESM-1b and BERT-RBP also provide the advantage of automatic feature generation without depending on human expertise or other tools to get a secondary structure or tertiary structure. It provides a generalizable solution that may be used for other biosequence-related tasks.

There are still some improvements that can be done in the generation of ESM/BERT features. While fine-tuning the BERT-RBP model, we used only the RNAs binding to the TIAL1 protein because of constraints in time and computing cost. TIAL1 protein is an mRNA binding protein with three RNA recognition motifs, and these motifs bind to adenine and uridine-rich regions of several mRNAs and pre-mRNAs. So, the model learned representations of adenine and uridine-rich motifs and may be better suited for mRNAs. As a result, the representations may not have generalized very well with our ncRNA dataset. We may expect better performance if we fine-tune the BERT-RBP model with more representative ncRNAs. Rather than using only one RBP complex for fine-tuning, we can choose several RBP complexes of several types. Nevertheless, comparable performance in current ESM/BERT features also indicates that the models learned features that might be universal to both coding and noncoding RNAs.

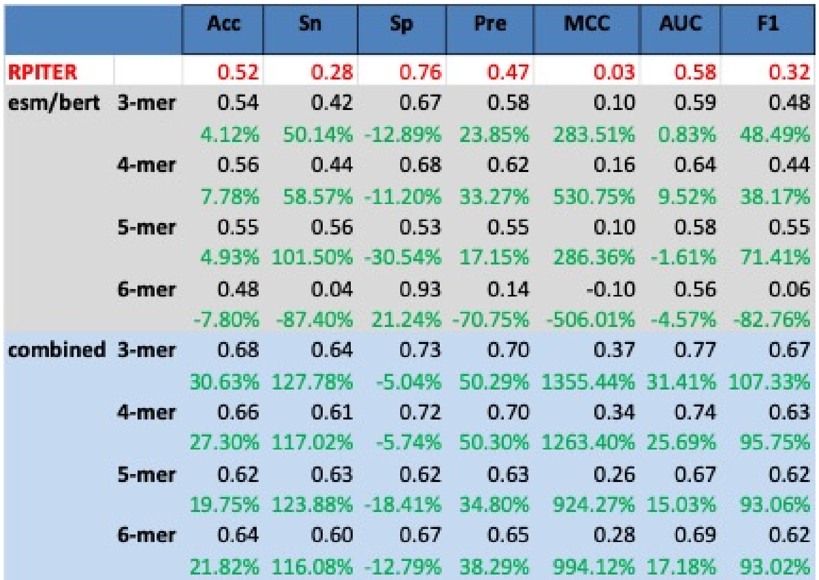
BERT-RBP fine-tuning was also done only 3 epochs because of time and computing cost constraints. At the end of fine-tuning, the accuracy was around 75%. With good computing resources and representative fine-tuning data, we can fine-tune for longer epochs until the accuracy increases. The representations produced would be more generalizable toward ncRNA sequences.

ESM just released their second version of the pre-trained model, which was trained on a larger amount of protein sequences. Using this version would be much recommended in the future experiments.

In summary, our experiments suggest that features from ESM-1b and BERT-RBP may be a good candidate in studying biosequences than manually collecting and creating features from secondary and tertiary structures. This allows an easier, faster, and automatic way of analyzing biosequences and creating classification models from them. We can improve the features by fine-tuning our BERT-RBP models with more representative RNA sequences and for a longer duration. The next consideration could be using the RPITER model instead of EDLMFC, to get better performance with RPITER features.



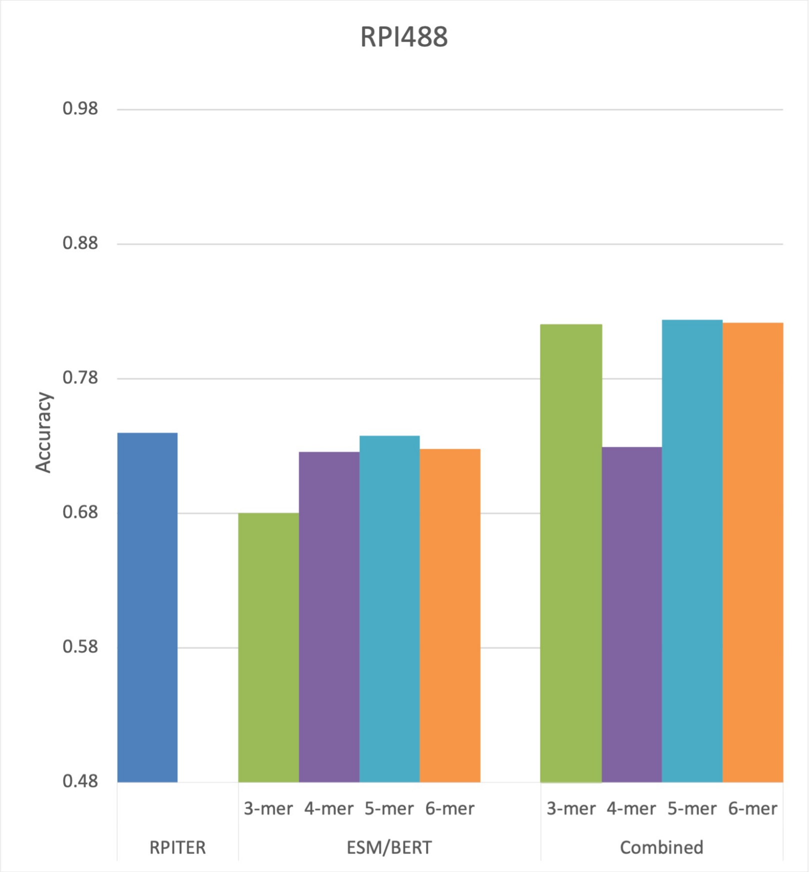
(a)



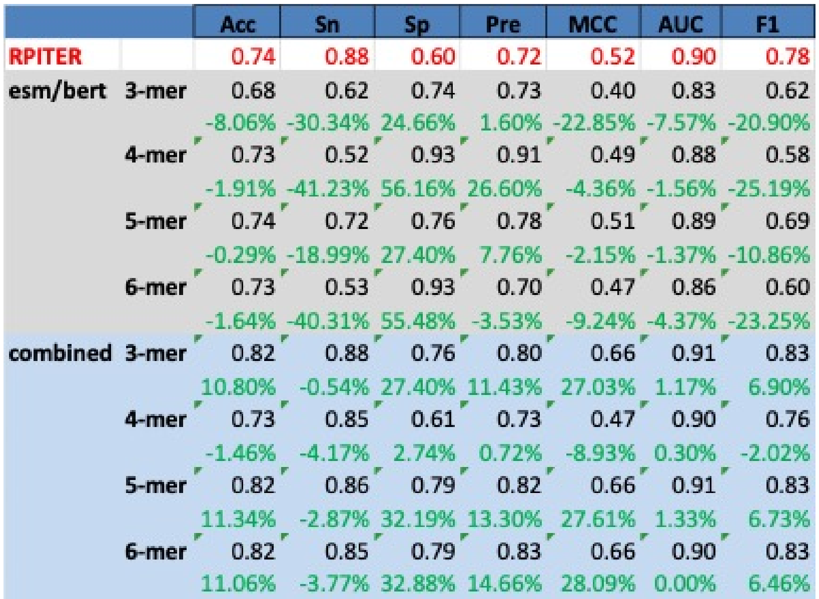
		Acc	Sn	Sp	Pre	MCC	AUC	F1
RPITER		0.52	0.28	0.76	0.47	0.03	0.58	0.32
esm/bert	3-mer	0.54	0.42	0.67	0.58	0.10	0.59	0.48
		4.12%	50.14%	-12.89%	23.85%	283.51%	0.83%	48.49%
	4-mer	0.56	0.44	0.68	0.62	0.16	0.64	0.44
		7.78%	58.57%	-11.20%	33.27%	530.75%	9.52%	38.17%
	5-mer	0.55	0.56	0.53	0.55	0.10	0.58	0.55
		4.93%	101.50%	-30.54%	17.15%	286.36%	-1.61%	71.41%
	6-mer	0.48	0.04	0.93	0.14	-0.10	0.56	0.06
		-7.80%	-87.40%	21.24%	-70.75%	-506.01%	-4.57%	-82.76%
combined	3-mer	0.68	0.64	0.73	0.70	0.37	0.77	0.67
		30.63%	127.78%	-5.04%	50.29%	1355.44%	31.41%	107.33%
	4-mer	0.66	0.61	0.72	0.70	0.34	0.74	0.63
		27.30%	117.02%	-5.74%	50.30%	1263.40%	25.69%	95.75%
	5-mer	0.62	0.63	0.62	0.63	0.26	0.67	0.62
		19.75%	123.88%	-18.41%	34.80%	924.27%	15.03%	93.06%
	6-mer	0.64	0.60	0.67	0.65	0.28	0.69	0.62
		21.82%	116.08%	-12.79%	38.29%	994.12%	17.18%	93.02%

(b)

Figure 4.1: **Performance in RPI369 dataset**: **(a)** Chart for comparison of accuracy across the feature sets, **(b)** Table for different performance metric values across the feature sets.



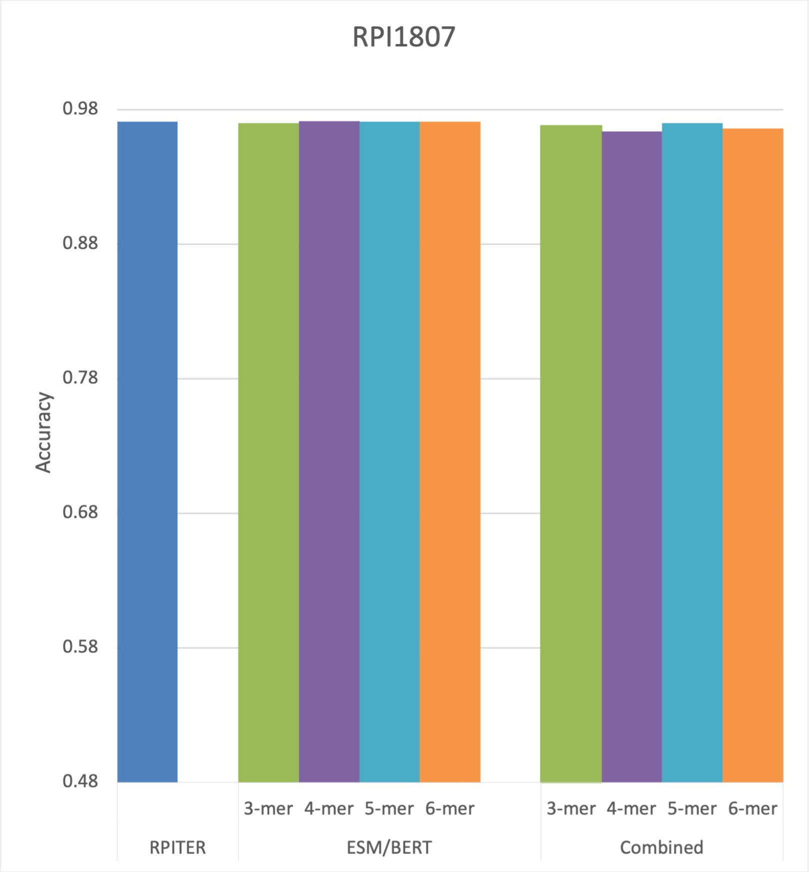
(a)



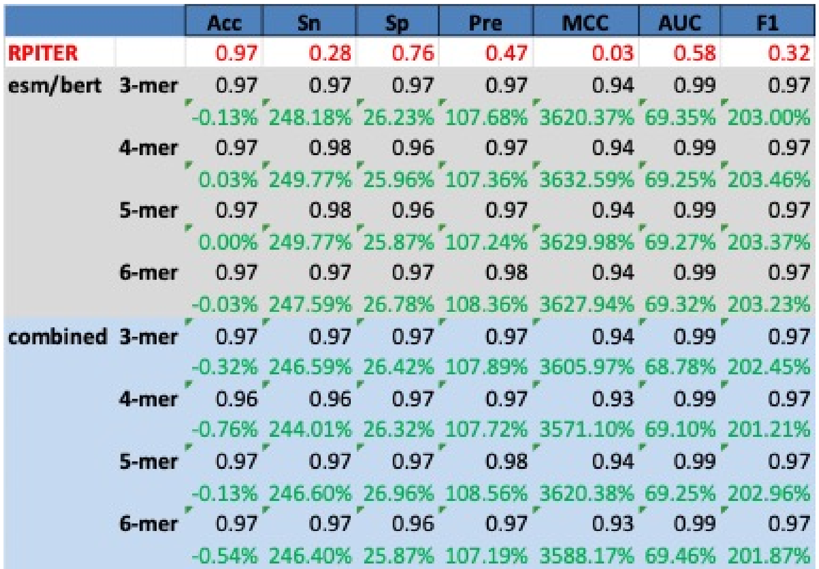
		Acc	Sn	Sp	Pre	MCC	AUC	F1
RPITER		0.74	0.88	0.60	0.72	0.52	0.90	0.78
esm/bert	3-mer	0.68	0.62	0.74	0.73	0.40	0.83	0.62
		-8.06%	-30.34%	24.66%	1.60%	-22.85%	-7.57%	-20.90%
	4-mer	0.73	0.52	0.93	0.91	0.49	0.88	0.58
		-1.91%	-41.23%	56.16%	26.60%	-4.36%	-1.56%	-25.19%
	5-mer	0.74	0.72	0.76	0.78	0.51	0.89	0.69
		-0.29%	-18.99%	27.40%	7.76%	-2.15%	-1.37%	-10.86%
	6-mer	0.73	0.53	0.93	0.70	0.47	0.86	0.60
		-1.64%	-40.31%	55.48%	-3.53%	-9.24%	-4.37%	-23.25%
combined	3-mer	0.82	0.88	0.76	0.80	0.66	0.91	0.83
		10.80%	-0.54%	27.40%	11.43%	27.03%	1.17%	6.90%
	4-mer	0.73	0.85	0.61	0.73	0.47	0.90	0.76
		-1.46%	-4.17%	2.74%	0.72%	-8.93%	0.30%	-2.02%
	5-mer	0.82	0.86	0.79	0.82	0.66	0.91	0.83
		11.34%	-2.87%	32.19%	13.30%	27.61%	1.33%	6.73%
	6-mer	0.82	0.85	0.79	0.83	0.66	0.90	0.83
		11.06%	-3.77%	32.88%	14.66%	28.09%	0.00%	6.46%

(b)

Figure 4.2: **Performance in RPI488 dataset**: **(a)** Chart for comparison of accuracy across the feature sets, **(b)** Table for different performance metric values across the feature sets.



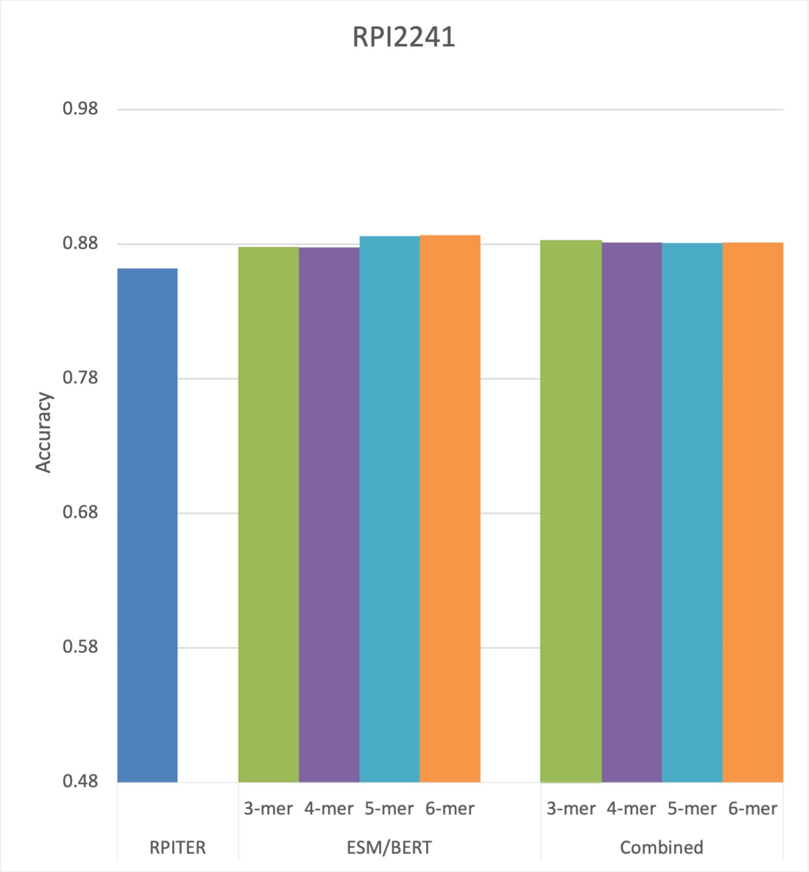
(a)



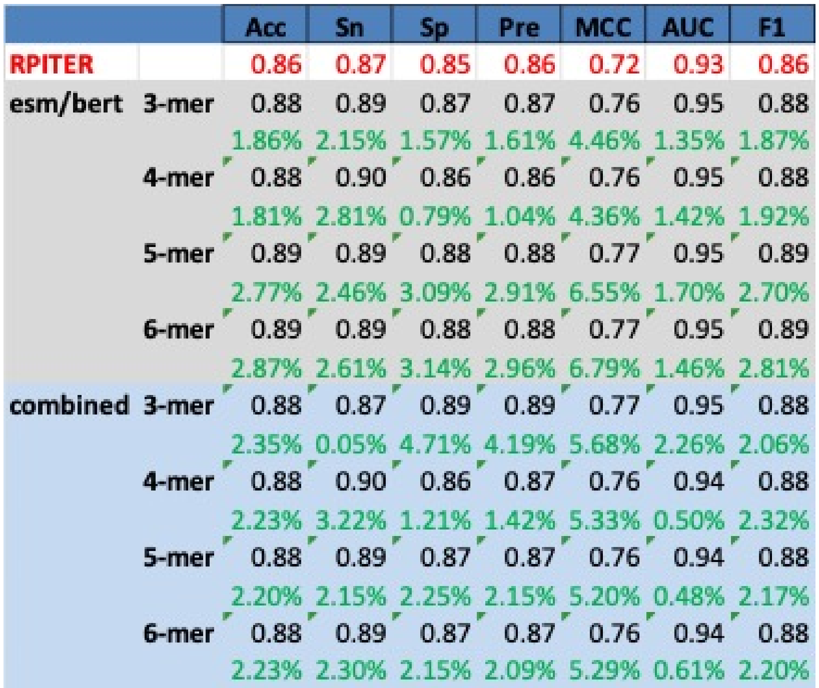
		Acc	Sn	Sp	Pre	MCC	AUC	F1
RPITER		0.97	0.28	0.76	0.47	0.03	0.58	0.32
esm/bert	3-mer	0.97	0.97	0.97	0.97	0.94	0.99	0.97
		-0.13%	248.18%	26.23%	107.68%	3620.37%	69.35%	203.00%
	4-mer	0.97	0.98	0.96	0.97	0.94	0.99	0.97
		0.03%	249.77%	25.96%	107.36%	3632.59%	69.25%	203.46%
	5-mer	0.97	0.98	0.96	0.97	0.94	0.99	0.97
		0.00%	249.77%	25.87%	107.24%	3629.98%	69.27%	203.37%
	6-mer	0.97	0.97	0.97	0.98	0.94	0.99	0.97
		-0.03%	247.59%	26.78%	108.36%	3627.94%	69.32%	203.23%
combined	3-mer	0.97	0.97	0.97	0.97	0.94	0.99	0.97
		-0.32%	246.59%	26.42%	107.89%	3605.97%	68.78%	202.45%
	4-mer	0.96	0.96	0.97	0.97	0.93	0.99	0.97
		-0.76%	244.01%	26.32%	107.72%	3571.10%	69.10%	201.21%
	5-mer	0.97	0.97	0.97	0.98	0.94	0.99	0.97
		-0.13%	246.60%	26.96%	108.56%	3620.38%	69.25%	202.96%
	6-mer	0.97	0.97	0.96	0.97	0.93	0.99	0.97
		-0.54%	246.40%	25.87%	107.19%	3588.17%	69.46%	201.87%

(b)

Figure 4.3: **Performance in RPI1807 dataset**: **(a)** Chart for comparison of accuracy across the feature sets, **(b)** Table for different performance metric values across the feature sets.



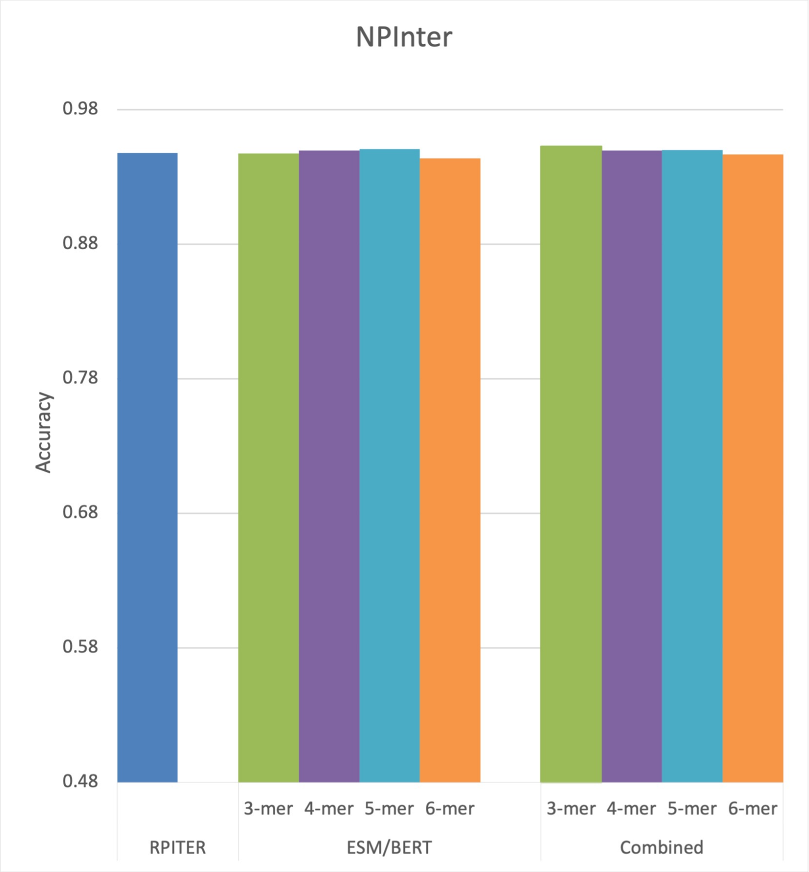
(a)



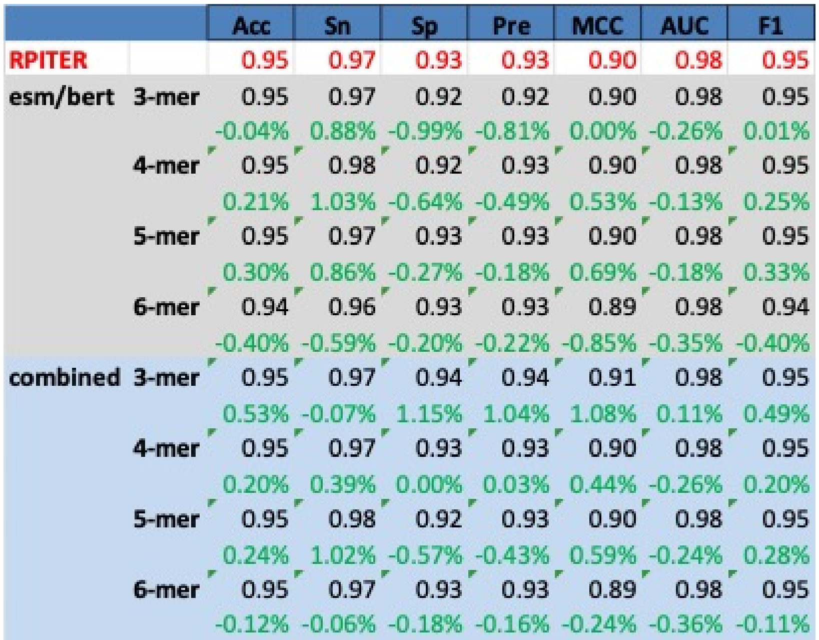
		Acc	Sn	Sp	Pre	MCC	AUC	F1
RPITER		0.86	0.87	0.85	0.86	0.72	0.93	0.86
esm/bert	3-mer	0.88	0.89	0.87	0.87	0.76	0.95	0.88
		1.86%	2.15%	1.57%	1.61%	4.46%	1.35%	1.87%
	4-mer	0.88	0.90	0.86	0.86	0.76	0.95	0.88
		1.81%	2.81%	0.79%	1.04%	4.36%	1.42%	1.92%
	5-mer	0.89	0.89	0.88	0.88	0.77	0.95	0.89
		2.77%	2.46%	3.09%	2.91%	6.55%	1.70%	2.70%
	6-mer	0.89	0.89	0.88	0.88	0.77	0.95	0.89
		2.87%	2.61%	3.14%	2.96%	6.79%	1.46%	2.81%
combined	3-mer	0.88	0.87	0.89	0.89	0.77	0.95	0.88
		2.35%	0.05%	4.71%	4.19%	5.68%	2.26%	2.06%
	4-mer	0.88	0.90	0.86	0.87	0.76	0.94	0.88
		2.23%	3.22%	1.21%	1.42%	5.33%	0.50%	2.32%
	5-mer	0.88	0.89	0.87	0.87	0.76	0.94	0.88
		2.20%	2.15%	2.25%	2.15%	5.20%	0.48%	2.17%
	6-mer	0.88	0.89	0.87	0.87	0.76	0.94	0.88
		2.23%	2.30%	2.15%	2.09%	5.29%	0.61%	2.20%

(b)

Figure 4.4: **Performance in RPI2241 dataset**: **(a)** Chart for comparison of accuracy across the feature sets, **(b)** Table for different performance metric values across the feature sets.



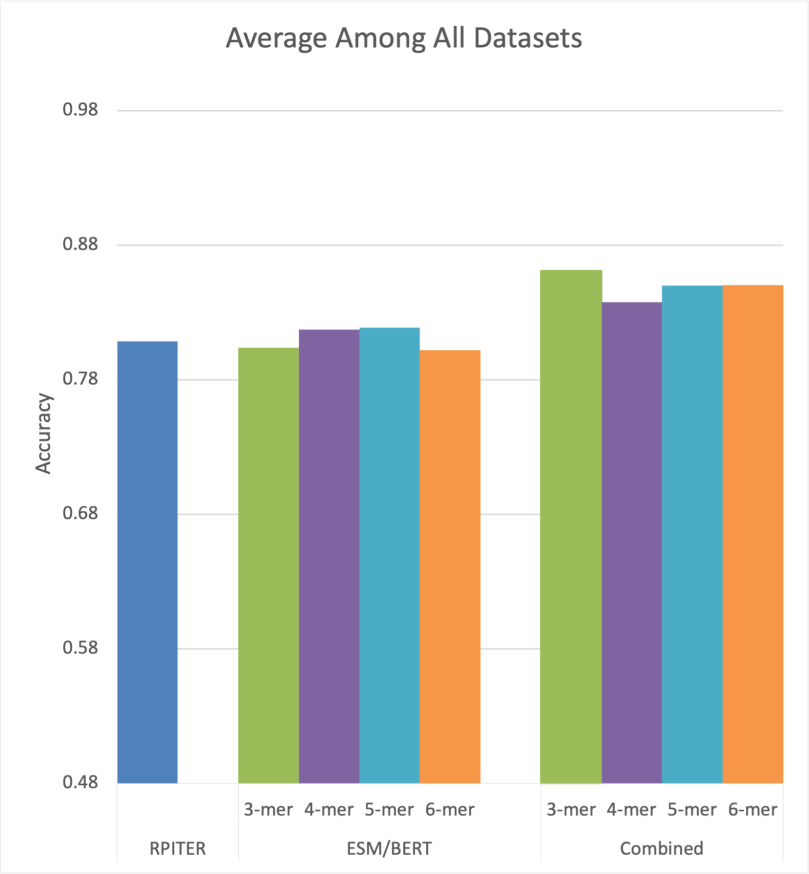
(a)



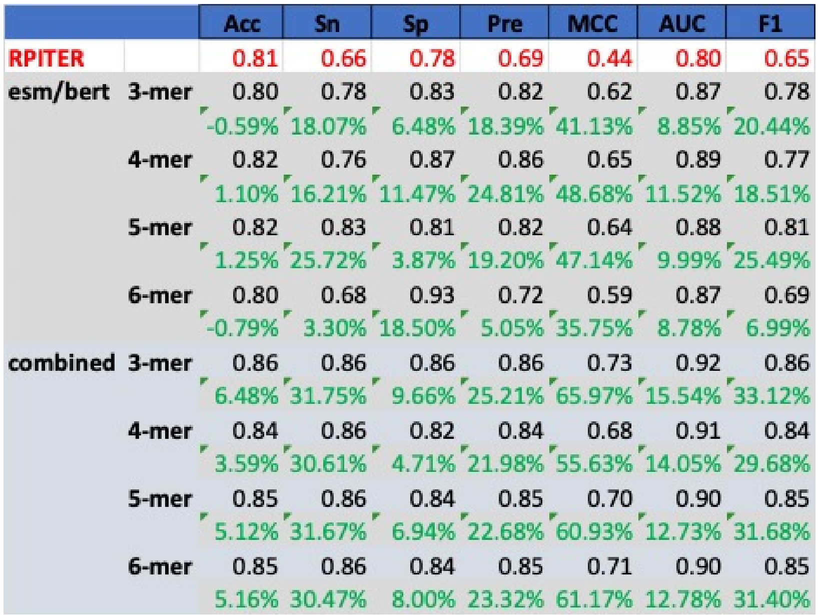
		Acc	Sn	Sp	Pre	MCC	AUC	F1
RPITER		0.95	0.97	0.93	0.93	0.90	0.98	0.95
esm/bert	3-mer	0.95	0.97	0.92	0.92	0.90	0.98	0.95
		-0.04%	0.88%	-0.99%	-0.81%	0.00%	-0.26%	0.01%
	4-mer	0.95	0.98	0.92	0.93	0.90	0.98	0.95
		0.21%	1.03%	-0.64%	-0.49%	0.53%	-0.13%	0.25%
	5-mer	0.95	0.97	0.93	0.93	0.90	0.98	0.95
		0.30%	0.86%	-0.27%	-0.18%	0.69%	-0.18%	0.33%
	6-mer	0.94	0.96	0.93	0.93	0.89	0.98	0.94
		-0.40%	-0.59%	-0.20%	-0.22%	-0.85%	-0.35%	-0.40%
combined	3-mer	0.95	0.97	0.94	0.94	0.91	0.98	0.95
		0.53%	-0.07%	1.15%	1.04%	1.08%	0.11%	0.49%
	4-mer	0.95	0.97	0.93	0.93	0.90	0.98	0.95
		0.20%	0.39%	0.00%	0.03%	0.44%	-0.26%	0.20%
	5-mer	0.95	0.98	0.92	0.93	0.90	0.98	0.95
		0.24%	1.02%	-0.57%	-0.43%	0.59%	-0.24%	0.28%
	6-mer	0.95	0.97	0.93	0.93	0.89	0.98	0.95
		-0.12%	-0.06%	-0.18%	-0.16%	-0.24%	-0.36%	-0.11%

(b)

Figure 4.5: **Performance in NPInter dataset**: **(a)** Chart for comparison of accuracy across the feature sets, **(b)** Table for different performance metric values across the feature sets.



(a)



		Acc	Sn	Sp	Pre	MCC	AUC	F1
RPITER		0.81	0.66	0.78	0.69	0.44	0.80	0.65
esm/bert	3-mer	0.80	0.78	0.83	0.82	0.62	0.87	0.78
		-0.59%	18.07%	6.48%	18.39%	41.13%	8.85%	20.44%
	4-mer	0.82	0.76	0.87	0.86	0.65	0.89	0.77
		1.10%	16.21%	11.47%	24.81%	48.68%	11.52%	18.51%
	5-mer	0.82	0.83	0.81	0.82	0.64	0.88	0.81
		1.25%	25.72%	3.87%	19.20%	47.14%	9.99%	25.49%
	6-mer	0.80	0.68	0.93	0.72	0.59	0.87	0.69
		-0.79%	3.30%	18.50%	5.05%	35.75%	8.78%	6.99%
combined	3-mer	0.86	0.86	0.86	0.86	0.73	0.92	0.86
		6.48%	31.75%	9.66%	25.21%	65.97%	15.54%	33.12%
	4-mer	0.84	0.86	0.82	0.84	0.68	0.91	0.84
		3.59%	30.61%	4.71%	21.98%	55.63%	14.05%	29.68%
	5-mer	0.85	0.86	0.84	0.85	0.70	0.90	0.85
		5.12%	31.67%	6.94%	22.68%	60.93%	12.73%	31.68%
	6-mer	0.85	0.86	0.84	0.85	0.71	0.90	0.85
		5.16%	30.47%	8.00%	23.32%	61.17%	12.78%	31.40%

(b)

Figure 4.6: **Average performance among all datasets**: **(a)** Chart for comparison of accuracy across the feature sets, **(b)** Table for different performance metric values across the feature sets.

Chapter 5

Conclusion

Features from BERT-RBP and ESM-1b have been shown to extract various information about RNA and protein from their sequences only. We tried to use the representations from these models as input vectors to an RNA-protein interaction prediction model with the hypothesis that the representations would capture the features that are necessary for RNA-protein interaction.

We used a state-of-the-art CNN-BLSTM model to test our features. Overall, ESM/BERT features provided comparable performance to the manually created RPITER features. The ESM/BERT features produced accuracies that were very close to the accuracy from RPITER features in all datasets except RPI369. In RPI369 dataset, ESM/BERT features showed increased accuracy of up to 7.8% as compared to that of the RPITER features. This purports the idea that ESM/BERT features as a suitable candidate for feature generation in RPI prediction. At the very least, the ESM/BERT features encode the information that has been encoded in RPITER features. In addition to its equal or better performance, the ESM/BERT features simplify, speed up and automate feature generation.

Although the combined features improved the accuracy of RPI for the most part, the improvements were most noticeable only in RPI369 and RPI488 datasets, the two small datasets. This suggests that the RPITER features, and ESM/BERT features encoded different information regarding RPI; hence, these features are additive in producing an improved performance. However, when the datasets are large, the combined features are similar to other features, indicating that RPITER features, and ESM/BERT features learned similar information from the RNA and protein. This also points to the idea of ESM/BERT features being a suitable candidate for RPI prediction.

In conclusion, these results provide support to the idea that these language-based models could provide a simple, automatic, and fast method to feature generation. The researcher can focus on model development rather than trying to come up with new features. At present, a lot of sequence data are available; however, their secondary and tertiary structure data are not readily available for all these sequences. In such cases, we can use ESM-1b and BERT-RBP, or other language-based models, to generate features from sequences only and use them for RPI classification or any other analytical tasks. ESM-1b and BERT-RBP, the Transformer models, improve with an increase in data that they are trained on. With new sequence data being created regularly, these models can produce more representative features, and, thus, help further in biosequence analytics.

Bibliography

[1] Xabier Agirre et al. “Long noncoding RNAs discriminate the stages and gene regulatory states of human humoral immune response”. In: *Nature communications* 10.1 (2019), pp. 1–16.

[2] Federico Agostini et al. “cat RAPID omics: a web server for large-scale prediction of protein–RNA interactions”. In: *Bioinformatics* 29.22 (2013), pp. 2928–2930.

[3] Krishnarao Appasani, Victor R Ambros, and Sidney Altman. *MicroRNAs: from basic science to disease biology*. Cambridge Univ Pr, 2008.

[4] Matteo Bellucci et al. “Predicting protein associations with long noncoding RNAs”. In: *Nature methods* 8.6 (2011), pp. 444–445.

[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[6] Helen M Berman et al. “The protein data bank”. In: *Nucleic acids research* 28.1 (2000), pp. 235–242.

[7] Pea Carninci et al. “The transcriptional landscape of the mammalian genome”. In: *science* 309.5740 (2005), pp. 1559–1563.

[8] Shuping Cheng et al. “DM-RPIs: Predicting ncRNA-protein interactions using stacked ensembling strategy”. In: *Computational biology and chemistry* 83 (2019), p. 107088.

[9] Francis Crick. “Central dogma of molecular biology”. In: *Nature* 227.5258 (1970), pp. 561–563.

[10] Fiona Cunningham et al. “Ensembl 2019”. In: *Nucleic acids research* 47.D1 (2019), pp. D745–D751.

[11] Qiguo Dai et al. “Construction of complex features for computational predicting ncRNA-protein interaction”. In: *Frontiers in Genetics* (2019), p. 18.

[12] Jennie Dusheck. “The Interpretation of Genes.” In: *Natural History* 111.8 (2002), pp. 52–59.

[13] Yanrong Ji et al. “DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome”. In: *Bioinformatics* 37.15 (2021), pp. 2112–2120.

[14] Yuening Jia. “Attention mechanism in machine translation”. In: *Journal of physics: conference series*. Vol. 1314. 1. IOP Publishing. 2019, p. 012186.

[15] Minna U Kaikkonen, Michael TY Lam, and Christopher K Glass. “Noncoding RNAs as regulators of gene expression and epigenetics”. In: *Cardiovascular research* 90.3 (2011), pp. 430–440.

[16] Ahmad M Khalil and John L Rinn. “RNA–protein interactions in human health and disease”. In: *Seminars in cell & developmental biology*. Vol. 22. 4. Elsevier. 2011, pp. 359–365.

[17] Stuart Knowling and Kevin V Morris. “Noncoding RNA and antisense RNA. Nature’s trash or treasure?” In: *Biochimie* 93.11 (2011), pp. 1922–1927.

[18] Benjamin A Lewis et al. “PRIDB: a protein–RNA interface database”. In: *Nucleic acids research* 39.suppl_1 (2010), pp. D277–D282.

[19] Ya-Pu Li et al. “A TRIM71 binding long noncoding RNA Trincr1 represses FGF/ERK signaling in embryonic stem cells”. In: *Nature communications* 10.1 (2019), pp. 1–13.

[20] Donny D Licatalosi, Xuan Ye, and Eckhard Jankowsky. “Approaches for measuring the dynamics of RNA–protein interactions”. In: *Wiley Interdisciplinary Reviews: RNA* 11.1 (2020), e1565.

[21] Minghui Liu et al. “HOTAIR, a long noncoding RNA, is a marker of abnormal cell cycle regulation in lung cancer”. In: *Cancer science* 109.9 (2018), pp. 2717–2733.

[22] Qiongshi Lu et al. “Computational prediction of associations between long noncoding RNAs and proteins”. In: *BMC genomics* 14.1 (2013), pp. 1–10.

[23] John S Mattick and Igor V Makunin. “Noncoding RNA”. In: *Human molecular genetics* 15.suppl_1 (2006), R17–R29.

[24] Usha K Muppirala, Vasant G Honavar, and Drena Dobbs. “Predicting RNA-protein interactions using only sequence information”. In: *BMC bioinformatics* 12.1 (2011), pp. 1–11.

[25] Xiaoyong Pan et al. “IPMiner: hidden ncRNA-protein interaction sequential pattern mining with stacked autoencoder for accurate computational prediction”. In: *BMC genomics* 17.1 (2016), pp. 1–14.

[26] Thomas D Pollard et al. *Cell biology E-book*. Elsevier Health Sciences, 2016.

[27] Muthukumar Ramanathan, Douglas F Porter, and Paul A Khavari. “Methods to study RNA–protein interactions”. In: *Nature methods* 16.3 (2019), pp. 225–234.

[28] Alexander Rives et al. “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences”. In: *Proceedings of the National Academy of Sciences* 118.15 (2021).

[29] Juwen Shen et al. “Predicting protein–protein interactions based only on sequences information”. In: *Proceedings of the National Academy of Sciences* 104.11 (2007), pp. 4337–4341.

[30] Richard Stefl, Lenka Skrisovska, and Frédéric H-T Allain. “RNA sequence-and shape-dependent recognition by proteins in the ribonucleoprotein particle”. In: *EMBO reports* 6.1 (2005), pp. 33–38.

[31] V Suresh et al. “RPI-Pred: predicting ncRNA-protein interaction using sequence and structural information”. In: *Nucleic acids research* 43.3 (2015), pp. 1370–1379.

[32] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[33] Jingjing Wang et al. “EDLMFC: an ensemble deep learning framework with multi-scale features combination for ncRNA–protein interaction prediction”. In: *BMC bioinformatics* 22.1 (2021), pp. 1–19.

[34] Ze-Lin Wang et al. “Comprehensive genomic characterization of RNA-binding proteins across human cancers”. In: *Cell reports* 22.1 (2018), pp. 286–298.

[35] Ying Wang et al. “De novo prediction of RNA–protein interactions from sequence information”. In: *Molecular BioSystems* 9.1 (2013), pp. 133–142.

[36] Keisuke Yamada and Michiaki Hamada. “Prediction of rna-protein interactions using a nucleotide language model”. In: *bioRxiv* (2021).

[37] Lin Zhu et al. “ChIP-PIT: Enhancing the analysis of ChIP-Seq data using convex-relaxed pair-wise interaction tensor decomposition”. In: *IEEE/ACM transactions on computational biology and bioinformatics* 13.1 (2015), pp. 55–63.

[38] Xueliang Zhu. “Seeing the yin and yang in cell biology”. In: *Molecular biology of the cell* 21.22 (2010), pp. 3827–3828.

Vita

Krishna Shah, born in Janakpur, Nepal, graduated with honors in BS in Biology program at the University of New Orleans (UNO) in 2017. After graduation, he worked in neuropharmacology lab for a couple of years and then started MS in Computer Science at UNO. The focus of his study is on Artificial Intelligence and Machine Learning. He has been working under the advisership of Dr. Md Tamjidul Hoque on a project combining biology and computer science knowledge. His research focuses on predicting the chance of physical interaction between protein and RNA given their sequences. Krishna has worked as an ML intern at Oracle Labs and is currently a software developer at Amazon Web Services.