Machine Learning Approaches for the Prediction of Binding Sites for RNA Binding Proteins

Mansha Imtiyaz

Master of Science

School of Computer Science
McGill University
Montreal, Quebec, Canada

July 30, 2018

A thesis submitted to McGill University in partial
fulfillment of the requirements of the degree of
Master of Science in Computer Science

Acknowledgements

I would like to thank my supervisor Prof. Mathieu Blanchette for introducing me to the field of bioinformatics, and I want to extend my gratitude to him for his unwavering support during my master's degree. He always directed me in the right direction. He was very patient in answering every single question I had and supported me through every challenge I faced. I also thank my co-supervisor Prof. Jérôme Waldispühl for guiding me throughout this past year. This thesis would not have been possible without his input and his guidance. I am blessed to have both of them as supervisors, they have always motivated and encouraged me through out the master program.

I would also like to express my gratitude to Faizy Ahsan, Atefah Mohajeri and Christopher Cameron for mentoring me and providing direction to my work. I am also grateful to Ozan Ciga, Ramchalam K R and Jaspal Singh for helpful discussion on my work.

I thank Prof. Mathieu Blanchette, Prof. Jérôme Waldispühl, School of Computer Science and Graduate and Postdoctoral Studies at McGill University for their financial support.

Abstract

RNA binding proteins (RBPs) play an essential role in many biological processes. Understanding the specific binding preferences of RBPs helps us in understanding the various steps of gene expression and may help in solving several genetic disorders. There are thousands of RBPs in humans, and only a small fraction of them are well understood. Current experimental methods for identifying RBP targets, such as CLIP-seq and RNAcompete, usually suffer from high false negative rate. In this work, we develop deep neural network models that allow us to learn binding preferences for a large number of RBPs from CLIP-seq data. We developed three deep architectures and used them to predict RNA-protein binding. We further analyze the importance of RNA secondary structure in RBP binding by incorporating computationally predicted secondary structure features as input to our models. We evaluate our model on the publicly available dataset of RBP binding sites derived from CLIP-seq. The results demonstrate that our approach achieves better or comparable performance with other state-of-the-art methods. Further, our model is able to automatically capture the interpretable binding motifs for several RBPs.

Abrégé

Les protéines de liaison à l'ARN (RBP) jouent un rôle essentiel dans de nombreux processus biologiques. Comprendre les préférences de liaison spécifiques de RBP nous aide à comprendre les différentes étapes de l'expression des gènes et peut aider à résoudre plusieurs troubles génétiques. Il y a des milliers de RBP chez les humains, et seulement une petite partie d'entre elles est bien comprise. Les méthodes expérimentales actuelles pour identifier des cibles de ces RBP, telles que CLIP-seq et RNAcompete, souffrent généralement d'un taux élevé de faux négatifs. Dans ce travail, nous développons des modèles de réseaux neuronaux profonds qui nous permettent d'apprendre les préférences de liaison pour un grand nombre de RBP à partir de données CLIP-seq. Nous avons développé trois architectures profondes et les utilisons pour prédire la liaison ARN-protéine. Nous analysons en outre l'importance de la structure secondaire de l'ARN dans la liaison RBP en incorporant des caractéristiques de structure secondaire prédites par des outils bioinformatiques en entrée de nos modèles. Nous évaluons notre modèle sur l'ensemble de données disponibles au public des sites de liaison RBP dérivés de CLIP-seq. Les résultats démontrent que notre approche atteint des performances meilleures ou comparables celles d'autres méthodes de pointe. En outre, notre modèle est capable de capturer automatiquement des motifs de liaison interprétables pour plusieurs RBP.

Contents

List of Figures

List of Tables

Chapter 1

Background

1.1. Molecular Biology

The flow of genetic information from DNA to RNA to protein is called the Central Dogma of Molecular Biology (see Figure 1.1). Watson and Crick discovered this dogma [1], which deals with the detailed residue by residue transfer of sequential information.

DNA and RNA are nucleic acids, made up of nucleotides, whereas proteins are made up of amino acids. Information is stored in DNA which in the process of *replication*, can be duplicated. In the *transcription* process, a stretch of DNA containing at least one gene can be copied into RNA, which is called messenger RNA (mRNA) if the gene encodes a protein. This means that mRNAs serve as genetic messengers as unlike DNA, which resides in the nucleus, they can move around the cell and carry instructions which can be used to synthesize a protein during the *translation* process. During protein synthesis, the sequence of an mRNA molecule is translated to a sequence of amino acids. The relation between the sequence of base pairs in a gene and the corresponding amino acid sequences is defined by the genetic code.

The process by which the information contained in a gene is read to synthesize a functional gene product, which can either be a protein or RNA is called *gene expression*. The process of gene expression involves transcription, i.e., the production of mRNA by the enzyme RNA polymerase, and the processing of the resulting mRNA molecule.

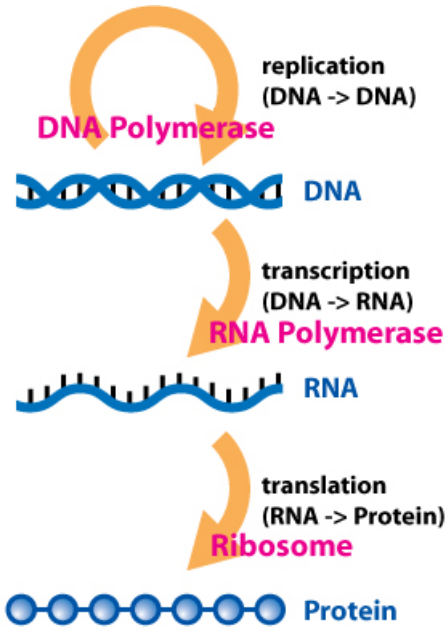

Figure 1.1: The central dogma of biology [3].

.

1.2. RNA Binding Proteins and their Roles

In order to decide what product needs to be created and in what amount, regulation of gene expression is necessary. To regulate transcription of nearby genes, transcription factors bind to regions of DNA. Once the DNA has been transcribed into RNA and before the translation happens, *post-transcriptional regulation* occurs at the RNA level. RNA-binding proteins (RBPs) take over about 10% of eukaryotic proteome with unique binding preferences and protein-protein interaction characteristics [4]. The remarkable diversity of RBPs allows for their utilization in numerous combinations, which gives rise to ribonucleoprotein complexes (complex of ribonucleic acid and RNA-binding protein), whose composition is unique to each mRNA. These RBPs influence the structure and interactions of RNA and regulate numerous post-transcriptional processes by regulating maturation, degradation, stability and transport of cellular RNAs [2].

The RBPs not only influence these processes but also provide a link between them [7]. Their proper functioning is necessary for complex post-transcriptional

events coordination (Figure 1.2). For instance, neuron-specific Nova proteins recognize intronic YCAY elements (Y = U/C) and thus, control the alternative splicing of pre-messenger RNAs [8]. SR proteins (proteins involved in RNA splicing which accompany the transcript through splicing process) such as SF2/ASF regulate translation initiation by enhancing phosphorylation of 4E-BP1 [9]. Similarly, miR369 recruits Argonaute2-FXR1 complex in dormant cells which were halted at G0/G1 phase and stimulates translation of TNF-alpha mRNA [10]. In 3'-UTRs, AU-rich elements (AREs) serve as docking sites for several proteins modulate mRNA stability [11]. ARE-binding proteins (ARE-BPs) from Hu family (HuB, HuC, HuD and HuR) generally stabilize the transcripts, while some ARE-BPs (TIA-1, AUF1) destabilize their target mRNAs [12].

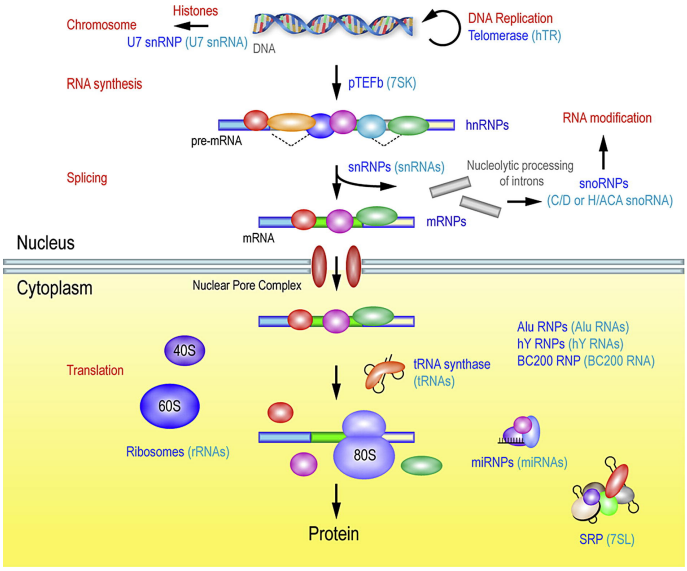

Figure 1.2: RNA-binding proteins (RBPs) regulate numerous post-transcriptional processes. Genetic information stored in chromosomal DNA is translated into proteins through mRNAs. In addition to the RBPs associated with mRNA, many different classes of RBPs interact with various small non-coding RNAs to form ribonucleoprotein (RNP) complexes that are actively involved in many different aspects of cell metabolism, such as DNA replication, expression of histone genes, regulation of transcription and translational control [5].

.

The effect on mRNA stability and translation rate can sometimes be co-related.

Though this is not always the case. For example, HuR enhances and CUGBP2 inhibits the translation of Cox-2 mRNA but both proteins enhance its stability [13]. Many RBPs are involved in more than one distinct process. For example, TIA-1 primarily destabilizes the target mRNAs but its presence also results in accumulation of stress granules in the cytoplasm which are formed when translation initiation is impaired [14].

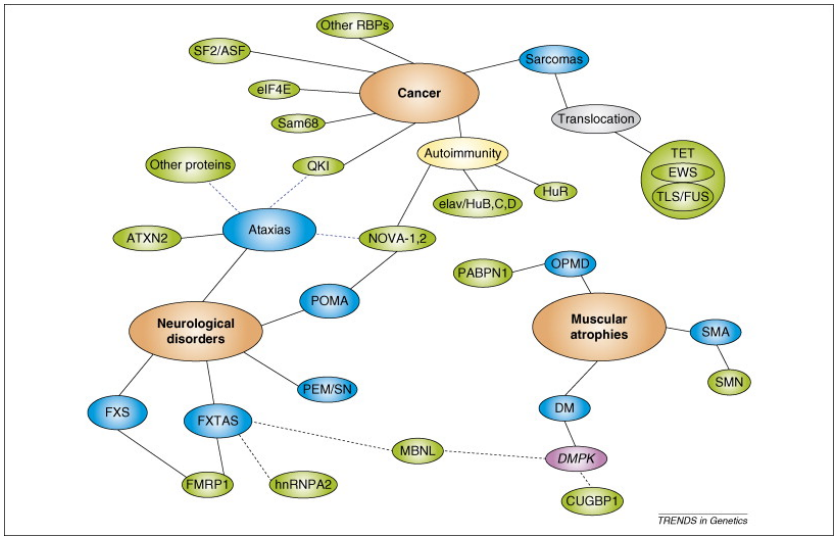

Figure 1.3: A network of RNA-binding proteins in human diseases [6].

.

1.2.1 RBP Mutations can cause Human Diseases

Due to their critical role in post-transcriptional regulation, many human diseases such as breast and lung cancer, muscular atrophies and neurological diseases are said to be caused in part due to mutations in RBPs or their binding sites [20]. As shown in Figure 1.3, aberrations in RBPs are directly or indirectly associated with specific diseases. For example, in 3' untranslated regions of the DMPK gene, the sequestration of RBPs CUG-BP1 and MBNL1 on trinucleotide repeats can result in myotonic dystrophy [15]. RBPs ATXN2, NOVA and QKI are associated with human inherited ataxias [16]. Mutation in RBPs is a common feature in various cancers. SF2/ASF and eIF4E are among a growing list of RBPs which have been characterized as oncogenes. NOVA and Hu proteins are associated with

brain tumors as they are targets of anti-neural antibodies. Understanding the binding preferences of RBPs helps us in understanding the molecular mechanisms of RBP mutations in disease which could lead to better-targeted therapies.

1.3. RNA Secondary Structure

RNA plays many roles in the storage and transmission of genetic information and exists in several forms, each with its own unique function. RNA is also an integral part of ribosomes, the site of protein synthesis, and some RNAs have been shown to have catalytic properties. Understanding the structure and function of RNA is important to a fundamental knowledge of genetics.

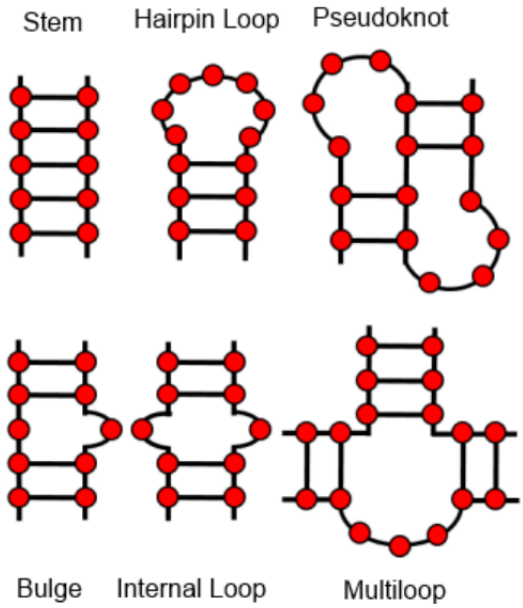

Figure 1.4: Different secondary structural motifs in RNAs are usually classified as stem, internal loop (or interior loop), multibranch loop. Figure taken from [23].

.

Single stranded RNA can fold back onto itself to form RNA secondary structure (Figure 1.4). Tertiary structure is then formed by higher order interactions after the formation of secondary structure. The secondary structure of an RNA sequence can be represented by assigning an index to each of the bases. In an

RNA molecule, the bases are indexed starting from the 5' end towards the 3' end. Assuming N is the length of sequence, we can define the RNA secondary structure S as a set of pairs $(i, j), 1 \leq i < j \leq N$ where each of the bases is paired with zero or one other base. Watson-Crick pairings A-U, G-C and wobble pairing G-U are the most common base pairs. A stack of such base-pairs is called a stem and loops are unpaired regions that are enclosed by base pairs. These loops, depending on the number of closing base pairs, can be called hairpin, internal, bulge or multi-loops (see Figure 1.4).

1.4. Binding Mechanisms of RBPs

The binding of RBPs to binding sites is determined by one or more RNA-binding domains (RBDs), of which there are more than 40 different types [22]. Some of the well-known RBDs are the RNA-recognition motif (RRM) [28], the heterogeneous nuclear (hn), RNP K homology (KH) [29], the double-stranded RBD (dsRBD) [30] and Pumilio (PUF repeats) homology (PUM-H) [31]. The architecture of RBPs is simple in the sense that they are constructed from individual RBDs that identify low affinity RNA stretches. Individual RBD sequences are short and thus, have limited ability to interact with RNAs on their own. By combining multiple domains with different arrangements and different number of copies, the versatility in specificity and affinity is achieved. The Pumilio family of proteins (Puf) is an example of such kind of RBPs with a C-terminal RBD that comprises of ∼40-aa long, consecutively arranged PUF-repeats, [31].

DNA-binding proteins recognize the sequence content of their binding sites but RBPs may recognize both sequence and structure of their binding sites. This is due to the different helical configurations in DNA and RNA. The B-form helix in DNA has a wide major groove which is easily accessible by proteins, whereas the A-form helix in RNA has a major groove that is too narrow and deep. The shape of RNA is a result of base-pairing between its nucleotides. As such, RBPs identify the regions where the major groove has been broadened by hairpins or bulges to recognize single stranded regions or the openings in double-stranded regions [24]. For instance, TAT, an HIV-1 protein, has high affinity to bind to three-nucleotide bulge loops [25], whereas Staufen, a Drosophila RBP binds without any sequence

preference to double stranded regions [26]. This shows that some RBPs bind to single stranded RNAs by directly reading out the primary sequence while others recognize the structure of the RNA or sometimes take into consideration both the sequence and the structure.

Thus, RBPs bind to a variety of structural and sequential contexts. However, it's not easy to know the binding preferences just by looking at their amino acid sequence. As such, a number of experimental methods have been developed in order to understand and investigate the binding preferences of RBPs, and they are discussed in the next section.

1.5. Experimental Methods for RBP Target Site Detection

Detection of RBP binding sites can be done using high-throughput or low-throughput techniques. Further, these experimental methods can be categorized into *in vivo* (if they are performed in living cells) or *in vitro* (performed in a controlled environment outside the cell).

1.5.1 *In vitro* Experiments

In-vitro methods allow to test a wide variety of binding sites by identifying the targets in non-biological conditions. Some of the examples are:

SELEX

Selection of ligands by exponential enrichment (SELEX) is an *in vitro* technique for identifying RBP binding targets. The process takes advantage of the fact that RNA binding proteins are capable of selecting their RNA ligands from large randomized pools of different RNAs [32]. Several "rounds" of experiments are performed, with each round consisting of the same sequence of steps (Figure 1.5). To synthesize the randomized RNA pool, DNA templates are constructed. The synthesized pool is then labeled and allowed to bind to the target RBP. The bound RNA is separated from unbound RNA and converted back into DNA tem-

plates. The process is repeated until at least 90% of the pool is bound to the protein of interest. At this point, these resulting RNAs are converted to DNA, cloned and sequenced [36]. One of the disadvantages of SELEX is that it doesn't always reflect the physiological binding sites as it only identifies high affinity RNA targets.

RNAcompete

RNAcompete is a high-throughput *in vitro* method for measuring the RNA sequence preferences of over 200 RBPs to more than 240,000 probe sequences which have been designed to cover each 9-mer at least 16 times [33]. It starts with generating an RNA pool which consists of k-mers in a variety of structural contexts, followed by pulldown of the RNAs bound to target RBPs. The name RNAcompete comes from the fact that individual RNA sequences compete to bind to proteins due to excess of concentrated RNA results. The recovered pulldown RNA is labelled and hybridized with complementary probes on a microarray. The final step is to measure binding affinity by computing the log-ratio between the recovered RNA in pulldown RNA population and the total pool signal.

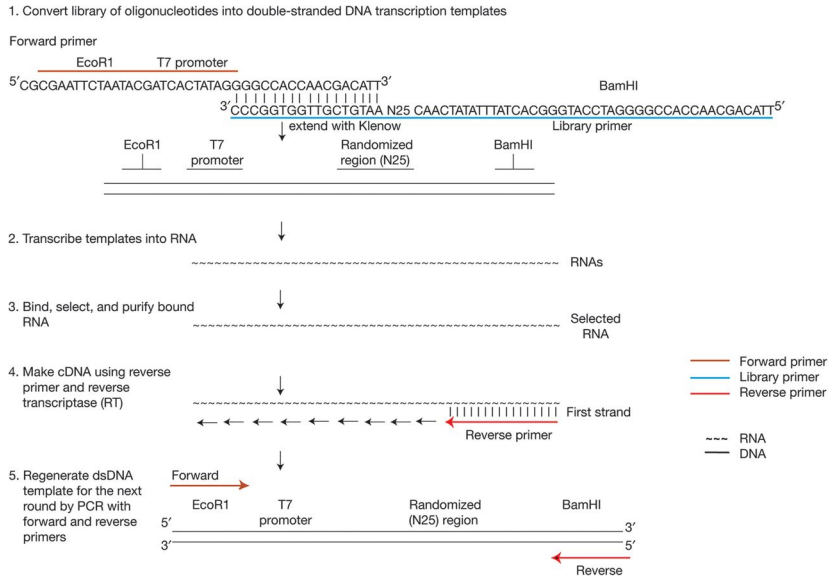

Figure 1.5: Schematic representation of a *round* of SELEX [36].

.

1.5.2 *In vivo* Experiments

Using *in vivo* methods such as HITS-CLIP, CLIP-seq and RIP-seq, one can query RBP-RNA interactions in biological conditions. The disadvantage is that they may identify both direct and indirect contacts and would require engineered cells to generate RBP levels that are much higher than normal, owing to noise and low resolution. The experiments are further complicated by the presence of other RBPs which could result in a competition or complex formation between them affecting the binding measurements.

RNA Immunoprecipitation

RNA immunoprecipitation (RIP) involves immunoprecipitation of a target RNA-binding protein (RBP) using an antibody. RNAs that are bound to the target RBP will be isolated during immunoprecipitation in non-stringent conditions and then sequenced. One drawback is its low specificity [34], which lead to the development of CLIP, explained below.

Cross-linking and Immunoprecipation

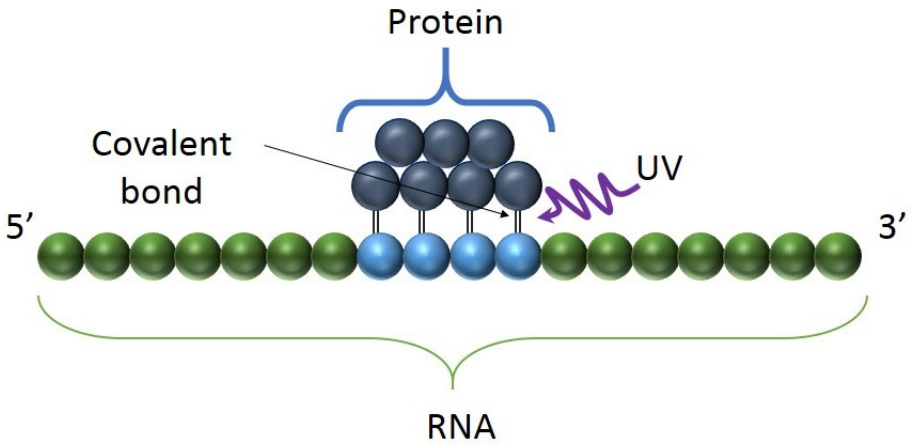

Figure 1.6: Covalent bonds are formed between RNA and proteins of interest on being exposed to UV light, followed by immunoprecipitation of RBP-RNA complexes [37].

.

One of the widely used experimental methods for identifying RBP binding targets are Cross-linking immunoprecipitation sequencing (CLIP-seq) protocols [35] (Figure 1.6). Covalent cross-links are formed between RBPs and bound RNA using UV radiation exposure on the cell or tissue culture, which is followed by immunoprecipitation of RBP-RNA complexes. The binding sites are then narrowed to a specific sequence length by partial RNrase digestion. After mapping to a reference genome, the binding sites are identified based on read profiles and the recovered RNA fragments are reverse transcribed into cDNA after proteinase K treatment [39]. This is followed by extraction and sequencing of cross-linked RNA fragments. Subsequently, the identified binding sites can be used to derive predictive models for binding motifs, and further to identify potential binding sites in new unlabeled RNA sequences.

More specialized techniques such as PAR-CLIP (photoactivatable-ribonucleoside-enhanced CLIP) improves cross-linking with photoreactive RNA nucleotides and can be used to locate binding sites at higher resolution [38]. This helps in decreasing signal-to-noise ratio over CLIP-seq. An alternative to high resolution CLIP method is iCLIP (individual-nucleotide CLIP) [43], which allows single nucleotide resolution of binding sites. eCLIP (enhanced CLIP) maps the binding sites of RBPs on their target RNAs using the iCLIP protocol and recently, has been widely promoted by the ENCODE consortium [40].

1.6. Limitations of Experimental Methods

Although there have been recent advances in experimental methods for the identification of RBP binding sites, there are several inherent limitations. Experimental methods require significant investment in time, work and effort, and as such are expensive and time-consuming. The data may contain many false positives due to inherent noise or contamination with non-cross-linked sites and a large number of binding sites may remain unidentified resulting in a high false negative rate. These limitations make the task of determining RBP target sites difficult. However, using these high-throughput technologies, a lot of RBP-related genome-wide data is being generated rapidly and deposited in databases such as the Protein-RNA interaction database (PRD). Section 1.7 details some of the

common databases of RNA-protein interactions and interfaces.

This data can serve as an important base for computational approaches which can be used to predict RBP-binding sites. Thus, the available high-throughput data acts as a gold standard for training and testing of less-expensive and faster computational prediction models, which are discussed in Section 1.8.

Table 1.1: Data sets commonly used for RNA-binding sites identification

Database	**URL**	**Description**
PDB [82]	http://www.rcsb.org/pdb	PDB contains data from experimentally determined 3D structures
NDB [75]	http://ndbserver.rutgers.edu/	The NDB contains information about experimentally-determined nucleic acids and complex assemblies.
CLIPdb [79]	http://postar.ncrnalab.org/	CLIPdb contains RBP binding sites from around 23 million experiments and 117 million predictions in the *mouse* and *human* transcriptomes. It provides various annotations (gene/RBP, molecular, etc.) for every transcript and its RBP target sites.
doRINA [85]	http://dorina.mdc-berlin.de/	doRINA database contains data for RNA-binding proteins and miRNAs. In the latest version available online, 136 RBP CLIP datasets have been used to identify RBP binding sites.
PRIDB [78]	http://pridb.gdcb.iastate.edu/	PRIDB contain data containing structural information for 926 RNA-Protein complexes. The data contains information about 1,475,774 amino acids from which 38% directly interact with RNA.

1.7. RNA-Protein Interaction Databases

This section lists some of the databases that focus on RNA-protein interactions, containing experimentally verified RBP target sites. These databases provide the users with the ability to search and browse known binding information and further, to find potential RNA-protein interactions. Table 1.1 provides URLs and descriptions for some common databases.

The Encyclopedia of DNA elements (ENCODE) [83] is a public research project with the aim to build a comprehensive list of functional elements in the human genome and also includes elements that act at the protein and RNA levels. The protein-RNA interaction database (PRD) contains data for 22 organisms such as human, Mus musculus and Drosophila melanogaster. It has 10817 interaction entries, referring to 1539 unique gene pairs [41]. Another widely used database is the RNA-associated interaction database (RAID) which has 1,208,008 entries for a total of 60 organisms [42].

The RNA-Binding Protein Database (RBPDB) [76] is a collection of RNA binding proteins and experimentally determined RNA binding specificities for RBPs of species such as Human, Mouse, and Drosophila melanogaster. It contains binding sites sequence logos for more than 70 Human RBPs and archives data from 14 types of RNA binding experiments. The databases consist of target site preferences for more than 200 RBPs in total, extracted from almost 1500 binding experiments [77]. For this thesis, Position Weight Matrices (PWM) and Position Frequency Matrices (PFM) for 71 Human RBPs were downloaded from this database for our experiments.

1.8. Computational Identification of RBP Binding Sites

As we saw in Section 1.6, the experimental methods used to detect RNA binding site proteins have certain limitations such as providing noisy measurements. The computational models take this data along with the observations as input. In the next sections, we present a review of computational models, starting from the analysis of the CLIP-Seq data to motif detection. Traditionally, computa-

tional approaches to predict binding sites represent motifs using a position weight matrix, which is explained in Section 1.8.2. Unlike DNA motif finding models, secondary structure of the binding sites has to be considered for predicting RNA binding sites.

1.8.1 Computational Analysis of CLIP-Seq Data

In section 1.5, experimental methods that generate RBP-bound RNA sequences have been detailed. The sequencing data produced from these experiments contain information about RBP binding sites on a transcriptome-wide scale. However, the data needs to be processed, filtered and analysed comprehensively in order to get useful biological insights.

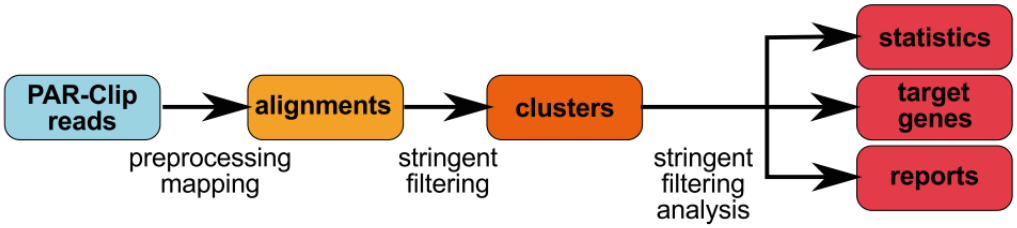

Figure 1.7: Computational pipeline for PAR-CLIP [86].

.

Figure 1.7 shows the computational pipeline for the analysis of PAR-CLIP [38] data. The first step is read mapping, where the reads are mapped to the genome and transcriptome. During processing, any experimental aids added to RNA fragments such as 3' adapter sequences are first removed. The reads are aligned and several of the short reads align to neighboring locations of the reference genome. The aligned reads are then grouped to create clusters. Clusters containing a single read are eliminated and the remaining clusters are annotated against a database of known transcripts. Quality scores are computed for each of the clusters based on the several parameters such as number of unique reads alignment, length of cluster, number of characteristic mismatches, etc. To mitigate the risks of having false-positive binding sites, other quality measures such as sense(coding strand direction) and anti-sense(non coding or template strand direction) alignment of

clusters are used. Anti-sense aligning of clusters are conservatively treated as false-positives though this may not always be the case if derived from unannotated anti-sense transcripts [86]. The false discovery rate is computed and the clusters are filtered by setting thresholds on their quality scores. The target genes associated with the clusters are then identified, as these clusters are annotated against known databases of transcripts. All the information related to the clusters (coordinates indicating start and end position, mapping the cluster to its location, the strand and other details) are stored in a BED [64] file. Figure 1.8 summarizes the different steps involved in the computational analysis of CLIP-seq data and the associated standalone programs designed for each step.

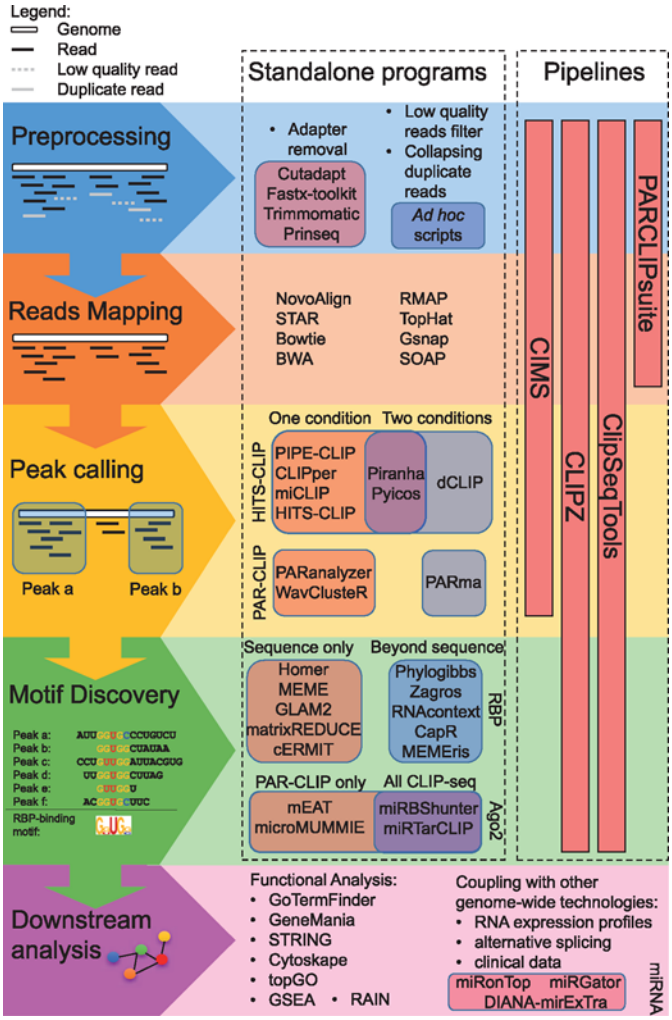

Figure 1.8: Recently, some computational pipelines have been developed which provide useful resources to deal with preprocessing steps, reads mapping, peak-calling procedure and other main steps of analysis [87].

.

1.8.2 Position Frequency Matrix (PFM) and Position Weight Matrix (PWM)

PWM and PFM are commonly used for representing motifs in biological sequences. These matrices indicate the probability of finding nucleotides at each position and as such, are used to capture the variable nature of binding sites.

```
Gene:  HuR-primary  Motif:  A.TGCACCC  Enrichment Score:  0.482942724828625
A:      0.219   1.364   -0.097  0.902   1.230   1.128   0.880   0.937   -0.311
C:      2.130   0.930   1.352   1.819   3.293   2.289   1.597   1.099   -0.252
G:      1.561   1.332   1.203   0.674   0.298   1.350   1.146   0.615   -0.557
T:      -0.523  -0.234  0.934   -0.009  -1.429  -1.369  -0.246  0.760   -0.573
```

Figure 1.9: Example PWM for HuR [74].

The RNA sequences are first aligned against each other and the number of times each nucleotide is found at a particular position is counted. This matrix of counts gives us a *Position Frequency Matrix (PFM)*. The frequencies in PFM are converted to normalized frequency values on a log-scale to get the corresponding *Position Weight Matrix (PWM)*, given by

$$W_{b,i} = log_2 \frac{p(b,i)}{p(b)} \tag{1.1}$$

where $W_{b,i}$ is the PWM value of base b in position i and $p(b)$ denotes the background probability of base b. $p(b,i)$ is the corrected probability of base b in position i and is given by

$$p(b,i) = \frac{f_{b,i} + s(b)}{N + \Sigma s(b')} \tag{1.2}$$

where N is the number of sites, $b' \in \{A, C, G, U\}$, $f_{b,i}$ gives the counts of base b in position i and $s(b)$ represents the pseudocount function that is used for probability correction for small samples sizes in order to have non-zero $p(b,i)$ values [89].

Figure 1.9 shows an example of HuR PWM. PWM computes probabilities at each position independent of other nucleotides in the motif and is used to identify candidate binding sites in new sequences. On scanning an RNA sequence S, each of the position j is scored as follows:

$$Score_j = \sum_{i=0}^{m-1} PWM_{S_{j+i,i}}, j \in 1, ...|S| - m \tag{1.3}$$

where S_{j+i} is the nucleotide at position $j+i$ in S, m is the length of the PWM matrix and $PWM_{b,y}$ is the value of nucleotide b at position y in the PWM, where $b \in \{A, C, G, U\}$. A threshold is used to filter the scores obtained using Eq. 1.3 to identify whether a binding site exists in a given sequence or not. There are several methods that are used to compute the PWM score as discussed in [93, 95].

1.8.3 Methods that use the Primary Sequence to identify RBP Target Sites

In the past, many motif-finding algorithms designed to analyze DNA sequences have been adapted to identify RBP binding specificities by scanning transcripts for potential binding sites. One such method is MatrixREDUCE [44] that infers RNA binding preferences from *in vitro* binding affinity data and associated labels. The model predicts the affinity associated with each binding site by representing it with a position-specific affinity matrix. Rather than taking a subset of sequences annotated as bound and unbound, the input for MatrixREDUCE is a set of quantitative values associated with each of the sequences.

Another popular motif discovery method, MEME (multiple expectation maximization for motif elicitation) [45] is a method used to find motifs in an unsupervised fashion but also able to use prior knowledge such as presence in all input sequences, length of a motif and whether it is a palindrome, when available. The algorithm has been designed to find ungapped, repeated sequence patterns in DNA or in proteins. The training set is a group of RNA or protein sequences and using statistical modeling preferences, MEME finds non-overlapping sets of approximately matching strings. MEME has been used to find mRNA targets in flies and yeast for Puf proteins [46].

Several other popular methods such as DRIMust [47] and Amadeus [48] have also been applied for analyzing RBP-bound RNAs. However, primary sequence motif-based models can incorrectly predict the binding preferences of RBPs as they miss important secondary structure context. For example, REFINE (relative filtering by nucleotide enrichment) is a method to find a group of consensus sequences from RIP-Chip datasets of yeast [49] but for Vts1p, it fails to identify known binding preferences from RIP-Chip data, whereas a method using motif finders that considers RNA accessibility can easily identify this primary sequence motif [50].

1.8.4 Prediction of RNA Secondary Structure

As seen in the above section, incorporating the RNA structure in order to predict binding sites can lead to improved results. However, we have to deter-

mine the structure first using experimental or computational methods. Among experimental methods, X-ray crystallography, cryo-electron microscopy and nuclear magnetic resonance (NMR) spectroscopy have been used [97, 101]. These methods, besides being time-consuming and difficult are sometimes impossible to use because of the large size and conformational flexibility of RNA structures.

Algorithms for RNA secondary structure prediction are usually based on the calculation of free energy using thermodynamic parameters from chemical experiments [51]. It is assumed that the RNA sequence folds into lowest free energy at equilibrium [52]. Consequently, the focus for secondary structure prediction is often on minimum free energy (MFE). A widely used program for secondary structure prediction is mfold [96]. It is based on a dynamic programming algorithm that uses energy parameters that take into account Watson-Crick and GU base pairs, various types of loops and terminal unpaired nucleotides and mismatches. Two of the most popular secondary structure prediction programs RNAfold [56] and RNAstructure [57] are also based on this principle and guarantee returning the lowest possible free energy structure.

However, the predicted MFE structure is not always biologically accurate as RNA secondary structure can fold into multiple structures in its lifetime and thermodynamic parameters may have substantial uncertainties [53]. Programs like Sfold [58] and RNAshapes [59] are some of the examples that find the optimal structure with the minimum free energy by narrowing the search to relatively few representative structures in large solution spaces. Such methods consider distributions of possible structures to find the secondary structure in the ensemble that best represents all the structures, such as the use of centroid structure [54]. In Sfold, sampling is based on the Boltzmann probability distribution and this can be used to produce centroid for each set of structures. The centroid structure has the minimum total base-pair distance to all other structures in the set. Using Sfold, multiple clusters and their centroids are produced from the ensemble and the centroid for the entire sampled ensemble is the ensemble centroid. This may be a more accurate representation of the correct structure than the lowest free energy structure and it may or may not represent the MFE structure. The centroids identified by Sfold are considered candidate structures and the base probabilities are computed from this representative sample. The RNAshapes algorithm cal-

culates shapes and their cumulative probabilities by independently enumerating the abstract shape space available to each sequence and then finds the thermodynamically optimal structure that has the common shape. There are five levels of abstraction in the current RNAShapes implementation and the folding space is partitioned into structural families that are represented by different shapes. The probabilities of all the structures assigned to same shape are summed to calculate the aggregate probabilities for shapes [59, 60].

The base-pairing probability pij that the ith and jth nucleotides in an RNA sequence form a base pair can be calculated by the McCaskill algorithm [80]. RNAfold uses a partition function to compute the probability for every possible pair [61]. A local-folding variant of RNAfold, RNAplfold considers only base pairs within a certain span [65]. In RNAplfold, average probabilities for base pair are calculated by considering windows of specific length containing the pair and averaging the probabilities. This approach has been shown to perform better in predicting mRNA secondary structure than the classical global folding algorithms [66] and is the method used for secondary structure prediction in this thesis.

1.8.5 Methods that incorporate Secondary Structure Information to identify RBP Target Sites

MEMERIS, an extension of MEME, was the first method to integrates structural information for searching motifs in a set of RNA sequences. For each *k-mer*, MEMERIS uses RNAfold to predict and precompute the probability that the word is in single stranded context. These values are then used as priors for possible motif start positions to guide the motif search. Thus, MEMERIS uses RNA accessibility information to guide and focus the search towards single-stranded regions. It has been used to find single stranded RBP motifs in biological sequence data such as SELEX [67].

Another method, RNAcontext [68], extends the accessibility information to include different types of unpaired regions such as external regions, bulges, hairpins, etc. RNAcontext was the first motif-finding algorithm used to find RBP binding preferences in multiple structural contexts. A set of RNA sequences is fed as input and their base preferences (given by PWM) and structural preferences as

computed by Sfold or RNAplfold are integrated in the input data. RNAcontext learns a model predicting input affinities and infers sequence and RNA structure preferences. This method was applied on an RNAcompete set of nine RBPS and outperformed both MatrixREDUCE and MEMERIS [68].

GraphProt [69] uses a graph-kernel strategy to integrate RNA sequence and secondary structural features to identify RBP binding sites. Secondary structures using RNAshapes are calculated and encoded as hypergraphs containing both sequential and secondary structure information. A dot-product function is used for similarity measure between graphs and features are extracted from the hypergraphs using graph-kernels. The input sequences are represented by over 30,000 features and a support vector machine (SVM) model is trained to identify the RBP sites using the extracted features.

One of the disadvantages of these methods is that observed data was used to construct the features for these models. The presence of frequent noise in the observed data may make the classifier learn the wrong underlying distribution which affects the final prediction. Choosing these features also requires considerable domain expertise and fine-tuning. These methods rely heavily on the choice of input features and may miss subtle features which are hidden in the input data and not explicitly encoded. Several deep-learning approaches have been developed in the past to address these challenges as they are data-driven and automatically learn high-level features. This approach is effective in integrating heterogeneous data and automatically learns complex patterns from multiple raw inputs. DeepBind was the first method that used deep neural networks to identify RNA binding sites from RNAcompete data [71]. This approach computes a binding score for each sequence by inputting a set of sequences in a Convolutional Neural Network (CNN) of 4 stages (convolution layer, rectified linear unit layer, pooling layer, non-linear neural net). However, Deepbind does not incorporate secondary structure information. Deepnet-rbp [70] was tested on CLIP-seq datasets and integrated k-mer frequency features of RNA sequences, secondary features using RNAShapes [59] and tertiary structure profiles using a deep belief network to identify RBP binding sites. They were the first to show that RBPs may have specific tertiary structure binding preferences.

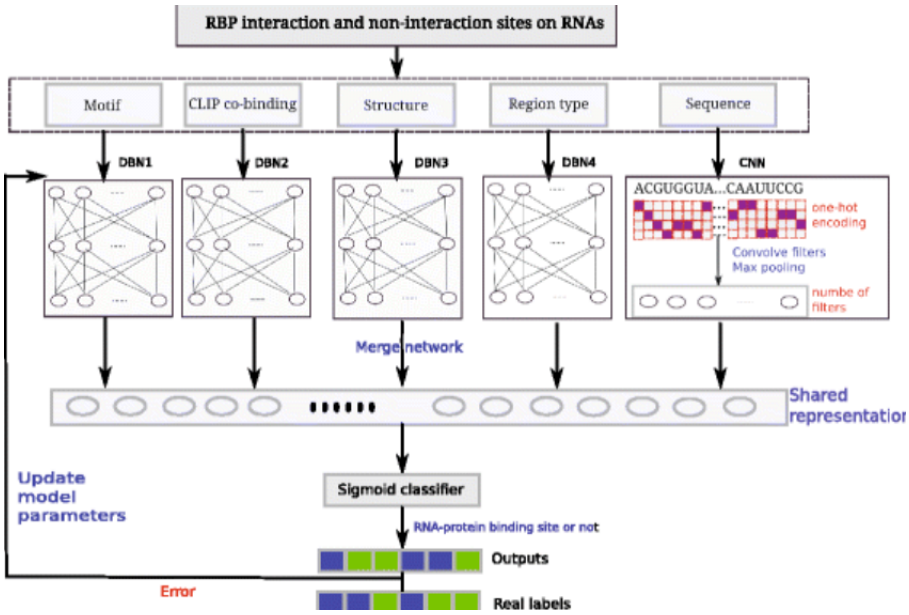

Figure 1.10: iDeep flowchart for predicting RNA-protein binding sites [72].

iDeep [72] uses sequential features using convolutional neural networks (CNNs) and deep belief networks (DBNs) to integrate different sources of data to identify RBP binding sites and sequence motifs. Each RNA sequence of window length 101 is encoded using 5 different feature sets including sequence, structure, region type, clip-cobinding and motif features. The probability of RNA secondary structure of each nucleotide is calculated using RNAplfold and Cluster-Buster [62] is used to score RNA sequences per 102 motifs obtained from CISBP-RNA [63]. Besides these, they also assign feature values based on regions (exon, intron, etc.). An additional layer is added to combine the output of multiple CNNs and DBNs, which are pre-trained independently during feature learning. Figure 1.10 shows the iDeep architecture.

A convolution layer tuned with trainable filters followed by a rectified linear ReLU and finally a pooling layer is used for the CNN architecture. The performance of iDeep was compared with other methods such as GraphProt and DeepBind and was found to yield best performance on 18 out of 31 experiments tested.

The reported accuracies of these tools range from 0.62 to 0.97 AUC [72]. For many RBPs, the performance is very far from perfect and there's still room for

improvement. We develop our methods with the aim to effectively boost the prediction performance using advanced deep learning techniques described in the next chapter.

Chapter 2

Overview of Machine Learning Approaches

In this chapter, we summarize the main machine learning approaches, explain what metrics are commonly used to measure performance of a machine learning algorithm and introduce deep learning concepts and methods.

2.1. Supervised Learning Algorithms

In supervised machine learning problems, the data consists of a pair of an input object (typically a feature vector) and a label (for classification problems) or a target real value (for regression problems) for each associated input. In classification problems, we typically have a positive and a negative class; e.g., in this case the positive class consists of sequences containing RBP binding sites identified by experimental methods as explained in section 1.5 and the negative class consists of sequences that are known to not bind to the RBP.

Typically the data is separated into a training set, which is used for training, and a smaller portion of the data, called the testing set. The test set is used to measure the accuracy of the model *(classifier)* built from the training set examples. The aim is to minimize the classification error on the test data set (i.e. predicted vs. true label difference). Many machine learning approaches can be used to generate the model. In the next sections, we will talk about performance metrics for machine learning models. We start by explaining the Random Forest Classifier

which has been successfully used in solving problems in the field of bio-informatics [90].

2.2. Performance Evaluation Metrics

The performance of a machine learning classifier can be measured using the four following metrics:

True positives (TP) is the number of positive examples that are correctly predicted as positives whereas *true negatives (TN)* is the number of negative examples that are correctly predicted as negatives. *False positives (FP)* is the number of negative examples incorrectly predicted as positives and *false negatives (FN)* is the number of positive examples incorrectly predicted as negatives.

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

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

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

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

$$F - Measure = \frac{2 * Precision * Sensitivity}{Precision + Sensitivity}$$

Performance metrics commonly calculated based on these metrics include sensitivity, specificity, precision, accuracy and F-measure. *Sensitivity* or *recall* is the fraction of true positives that are predicted to be positives. It is used to measure the ability to identify positive examples by a classifier. *Specificity* is the fraction of true negatives that are correctly predicted negative. It is used to measure the classifier's ability to identify negative examples. *Precision* is the fraction of predicted positives that are true positives. *Accuracy* is the percentage of the correctly predicted positive and negative examples. *F-Measure* is the harmonic mean of precision and recall.

Receiving Operating Characteristic (ROC) curve is a sensitivity vs.

specificity plot (rate of true positives to rate of true negatives), used for performance measurement of binary classifiers. The area under ROC curve (AUC) is widely used in machine learning classifiers for comparing performance. Higher AUC ($value > 0.5$) means better performing classifier.

2.3. Bias vs. Variance

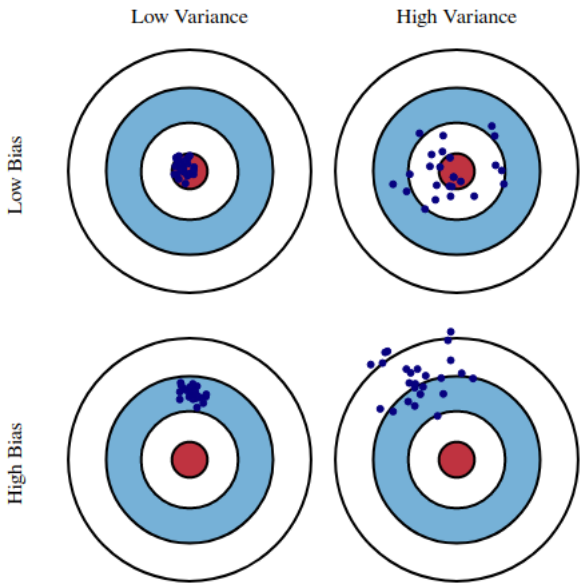

Figure 2.1: Four different cases plotted, representing combinations of both high and low bias and variance [103].

.

In supervised learning algorithms, classification errors are due to three sources: bias, variance and noise. If we repeat the entire model building process and gather new data each time, the resulting models will have a range of predictions due to randomness (noise) in the underlying data sets. Bias measures the difference between the average prediction of these models and the true value that we are trying to predict. Variance measures how much the predictions for a given point may vary between different realizations of the model. It occurs when our classifier

is too sensitive to the training set and any small changes in data may lead to big changes in model.

Considering Figure 2.1, we can explain bias and variance graphically using a bulls-eye diagram. Suppose different circles represent different models and target center is a model which correctly predicts the values for our input and as we move away from the center, the predictions get worse. We try this model building process with different realizations of our data. The predictions are sometimes closer to the target center indicating a good distribution of training data, while if our training data has outliers, the predictions will be away from the bulls-eye.

From the diagram, we can see that the best combination is low bias and low variance, which in practical cases, isn't always possible. However, by varying the complexity of the model, bias and variance can be traded off.

In terms of hypothesis space, not having good enough hypotheses in the considered class or in other words, restricting the hypothesis space results in *bias*. This moves the fit towards a simpler model and away from the best possible fit of the training data, which can result in *under-fitting*. Under-fitting can be avoided by including more learning parameters in the model, to make sure not to ignore relevant patterns existing in the data. On the other hand, *variance* occurs when the hypothesis class contains too many hypotheses. This can result in *over-fitting*. It occurs when there are too many learning parameters and the model instead of learning the underlying patterns, ends up learning random noise in the data.

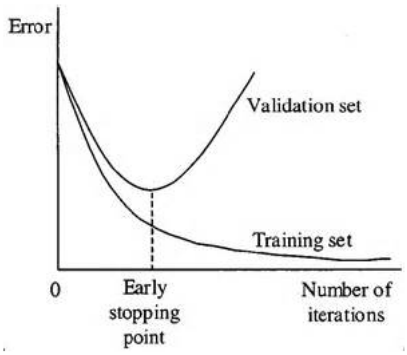

Figure 2.2: Early Stopping after a certain number of iterations can help prevent overfitting [104].

.

Regularization and early stopping are some of the techniques that have been used to avoid over-fitting. *Regularization* describes a broad of range of techniques used to limit the complexity of the model. Depending on the type of learner, regularization can refer to pruning a decision tree, adding some penalty to the cost function in regression or using dropout in neural networks. In decision trees, regularization is done by setting a stopping criteria for further splitting the node (e.g. minimum gain, maximum depth, etc.). In regression, if you have a large number of features, a squared magnitude (Ridge Regression) or an absolute value of magnitude (Lasso Regression) of coefficient is added as a penalty to decrease the model complexity.

Another technique is *Early-Stopping*, which means stopping the training process after a certain number of iterations, before the model begins overfitting. With each iteration, the training model improves. However, after a point the models stops generalizing and starts learning the noise in data, which can be avoided using early stopping.

2.4. Hyperparameters and Cross-validation

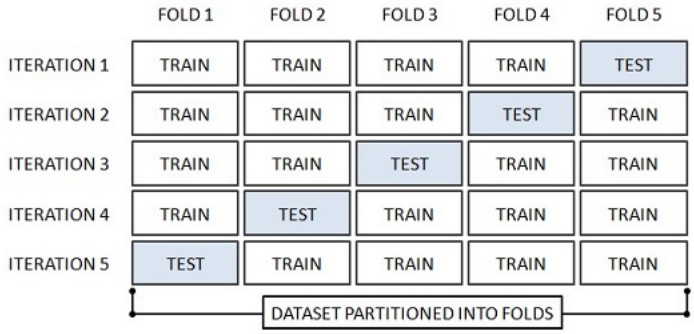

Figure 2.3: 5-fold Cross Validation [92].

.

A *hyperparameter* is a parameter that is selected and optimized before the training of the machine learning model. They represent 'high-level' properties of a model such as model complexity or how fast it should learn. For example, the learning rate of a feed-forward neural network, batch sizes, momentum parameters

are all hyperparameters that are set before the training begins. In contrast, the weights of the convolutional and fully connected layers are trainable and optimized during the training. Although there are methods for optimizing and automating the hyperparameter search, hyperparameter adjustment usually relies on manual engineering and it is crucial for converging to a good optimum.

Cross-validation is used for measuring the performance of the model and to assess how they perform on a dataset outside the training data, called the test set. This helps us to get a more reliable estimate of the training error when the available data is not large enough to allow a partition for the test set. In k-fold cross-validation (Figure 2.3), the data is divided into k-subsets and the algorithm is then iteratively trained on k-1 folds while the left-out fold is used as the test set. This allows us to use the original training set to tune the hyperparameters and the unseen test set can be used to select the final model. Another variation is leave-one-out cross-validation (LOOCV), in which one observation is left out at each step.

2.5. Random Forest Classifiers

A random forest (RF) classifier (Figure 2.4) is an ensemble algorithm, which creates a set of decision trees from randomly selected subsets of the training set. From a set of m variables selected at random from the input features set, the best split is chosen at each node of the tree. The weighted votes from different decision trees are then combined and the majority vote gives us the final class of the test object. An RF algorithm easily adapts to correlation and interaction among features. Due to this reason, RF is widely used in high-dimensional genomic data analysis [91]. For some number of trees T, the Random Forest classifier is trained as below [105].

For t=1 to T:

1. Sample N cases at random with replacement to create a subset of the data.

2. Repeat at each node d or until the stopping criteria is reached:

   (a) From m features in total, randomly select a subset of k features , where $k << m$

(b) Choose the best split point from among these k features

(c) Split d into daughter nodes using the best split

Random Forest creates random subsets of the features and builds trees from these subsets. This may help in preventing over-fitting, but can make computation slower depending on the number of trees.

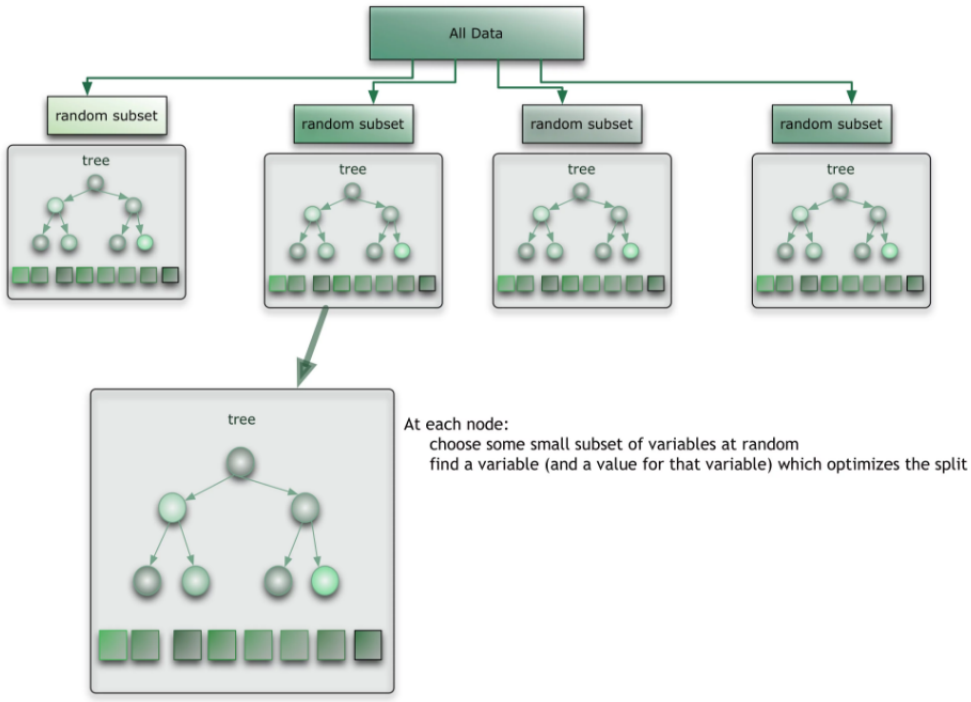

Figure 2.4: Illustration of a random forest algorithm [94].

.

2.6. Deep Learning

In traditional machine learning approaches, feature vectors are used as inputs and as such, these methods rely heavily on the choice of features, or data representation. Choosing these input features requires considerable domain knowledge and expertise and a good set of features can increase the accuracy of the classifier substantially. However, it may be possible to design better features in terms of

the objective function and as such these features may not be optimal. In *Representation Learning*, instead of designing these features by hand, we let the model learn them on its own. An example of such a model is a neural network explained in Section 2.6.1. If an image is given as an input to a neural network, it outputs a vector which may be the feature representation of the image. Here, the neural network will be called a representation learner. This can be followed by another neural network which can act as a regressor or a classifier for prediction models. Deep learning is representation learning that combines several non-linear transformations to learn multiple levels of representation. Stacking these different layers of transformation one on another helps in extracting underlying features hidden in the data. In lower layers, the features encode several low-level features such as the edges of an input image while higher layers extend on top of lower layers to represent abstract features such as faces, contours, etc. This enables the model to generalize to new combinations of learned features besides those seen during the training [106].

2.6.1 Feed Forward Neural Networks

Feed-forward neural networks or deep feed-forward networks form the basis of many neural networks used in the recent times. In a feed-forward neural network, the goal is to approximate some function f. Let a classifier map an input x to category y, shown as $y = f(x)$. A mapping $f(x; \theta)$ is defined and a feed-forward neural network learns the value of the parameter θ which results in the best function approximation [107].

As shown in Figure 2.5, a weighted linear combination of inputs or outputs from previous layers is fed into a neuron. A non-linearity function is applied and the output produced is forwarded to the next layer sequentially (forward propagation). These neurons are called hidden as they can model hidden variables within the data. The connections between the neurons do not form cycles as there are no feedback loops from the output neuron to itself. If extended to include feedback connections, the network is called recurrent neural network, which will be explained in Section 2.7.

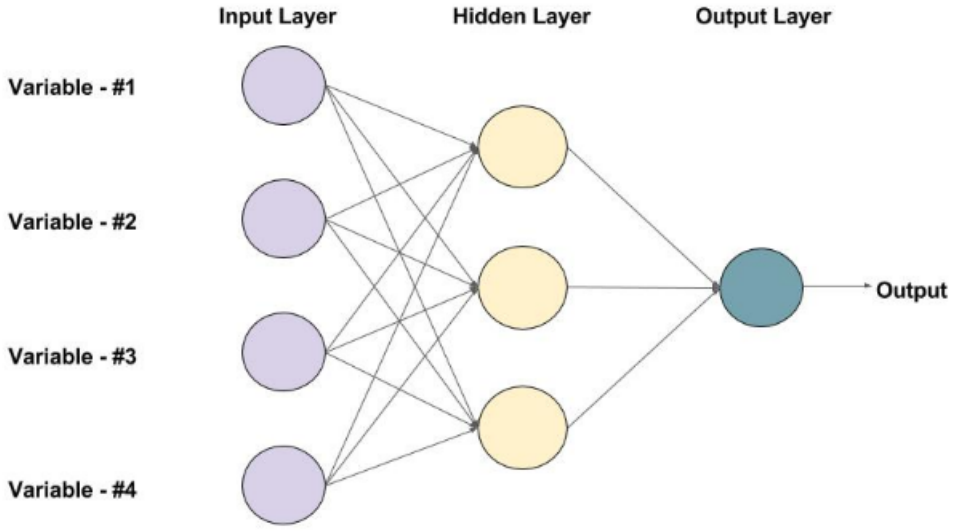

Figure 2.5: Feed-forward neural network illustrated where each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another [110].

.

Backpropagation is an algorithm for supervised learning of artificial neural networks using gradient descent and is used in order to learn the weights of the connections. The gradient of the cost function with respect to the weights is computed using the back propagation algorithm [108]. Batches of data are iteratively passed through the network and the weights are updated so that the error is decreased. The learning rate is one of the hyperparameters which determines the amount by which the weights are changed.

2.6.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are comprised of one or more convolutional layers followed by fully connected layers, whereas in a multi-layer feed-forward neural network, all layers are fully connected. Feed-forward neural networks do not consider the spatial structure of the input. The spatially far apart pixels are treated the same as the pixels that are close together. A CNN takes advantage of the 2D structure of input image or other 2D input. CNNs are very useful for extracting information from data that contain local groups which

are correlated and form distinct patterns (motifs). The main function of a convolutional layer is to extract features from the input image. The filters act as feature detectors. A feature map is produced by sliding the filter over the image and computing the dot product at each location. The convolution output is transformed by a nonlinearity such as a rectified linear unit (ReLU) before being fed into the next layer. ReLU allows the smooth propagation of gradients between layers in deep architectures without compressing the input. It is defined as $ReLU(x) = max(0, x)$.

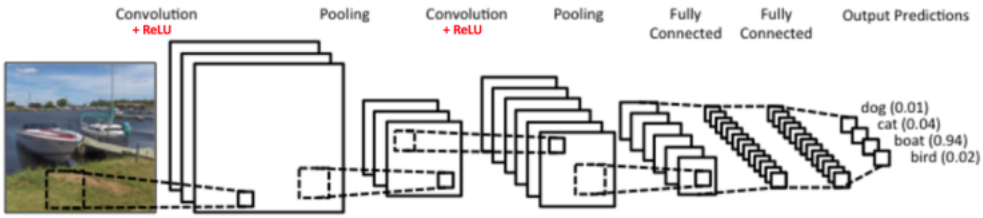

Figure 2.6: A simple convolutional neural network [111].

.

CNNs are invariant to shifting of local patterns/motifs along the input. The same set of filters are used for all positions in the input. Usually the convolutions are coupled with pooling layers that take the maximum, sum or the average of output feature maps. This reduces the dimensionality while still retaining the most important information. The output from these layers represent high-level features of the input image. The last convolutional layer is usually followed by one or more fully connected layers. These layers are used for classification from the extracted features. The classification process is guided by supervised training. Figure 2.6 shows an example of a generic Convolutional Neural Network.

In the past, Xavier initialization [119] has been used for initialization of the filters in a CNN with the assumption that there is no non-linearity between the layers. The variance of each neuron among the layers is kept the same using this initialization. However, Xavier initialization doesn't give excellent results when using ReLU nonlinearity as it halves the output variance by killing half the distribution. This was extended to ReLU nonlinearity by introducing layer sequential uniform variance - LSUV initialization [120]. The weights are pre-

initialized and the output variance of each layer is normalized. LSUV initialization is very fast and allows for learning of very deep nets.

CNNs are mostly used for image classification tasks [112, 113]. Recently, they have have also been used in the classification and identification of motifs in genomic sequences [61, 71]. DNA/RNA sequences can be encoded as one-hot-encoding as they are discrete sequences. One-hot-encoding takes a sequence of length k and encodes it into a matrix of dimension k x 4. Each column represents one of the four nucleotide bases (A,C,G,T/U). If the position contains the base corresponding to the column, the entry is set to 1, otherwise 0.

2.7. Recurrent Neural Networks

Recurrent Neural networks (RNNs) are networks used for processing sequential data such as text or genomic sequences. In feed-forward neural networks, the only input considered is the current example, whereas in recurrent neural networks, the input is not only the current input example but also a summary of what has been perceived previously in time.

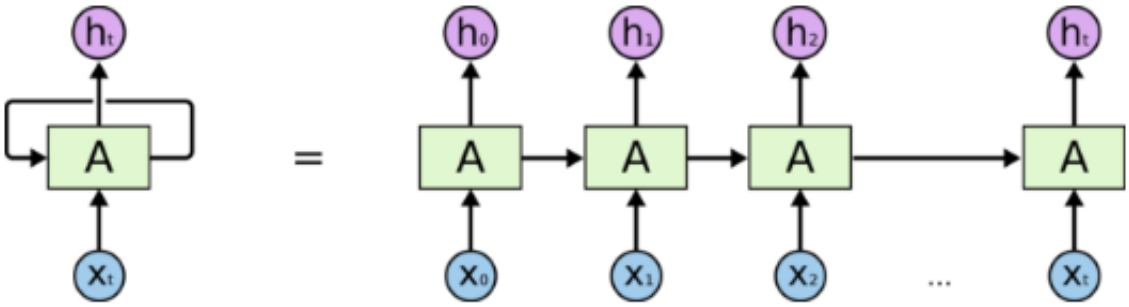

Figure 2.7: A part of recurrent neural network, A, looks at input X_t and outputs a value h_t. The loops pass information from one step of the network to another. The diagram shows the loops unrolled in time [114].

.

This makes RNNs great for processing sequential data. This is achieved by parameter sharing across different time steps, which enables them to share useful representations between them. Each input is processed one element at a time and

a state vector is maintained that contains a summary of all observations seen so far. At each time step, the hidden state is updated using the input and the output produced is the non-linear combination of the input and state from previous time steps. Figure 2.7 shows an RNN unfolded in time. Unfolded RNNs are quite similar to deep feed-forward networks, with weights restricted to be the same across time points.

A common issue that arises in training RNNs on long sequences is that RNNs have difficulties in learning the interactions between inputs that are several steps apart. The recurrent layer multiplies each input x_t with the weights w_j at each time step. When weights $w_j > 1$, the gradients explode, i.e. the gradient tends to go to ∞ as time increases. When $w_j < 1$, the gradient decays exponentially to 0. This causes the network to not learn long range dependencies that are temporally far apart [116].

2.7.1 Long Short Term Memory

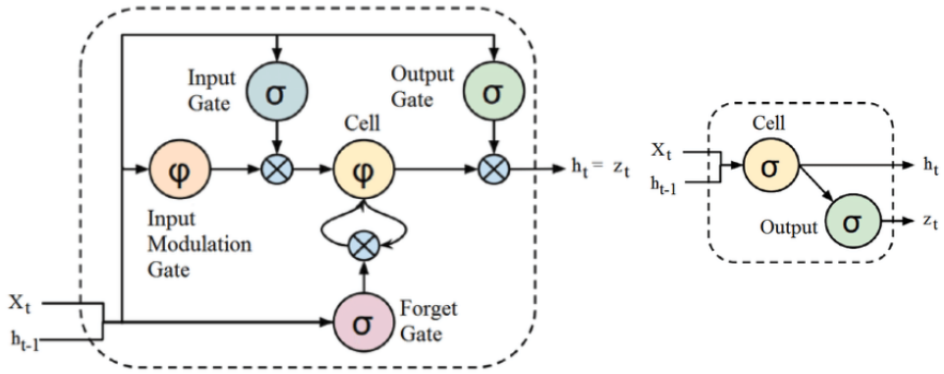

Figure 2.8: LSTM (left) replaces the normal RNN cell (right) and uses the input, forget, output and save gate. Figure taken from [117].

To solve the problem of vanishing gradient in RNNs, recurrent architectures with gating mechanisms were introduced, such as Long Short term Memory (LSTM) [115]. As shown in the Figure 2.8, LSTM replaces the normal RNN cell with recurrent units that have an explicit memory cell whose natural behavior is to remember inputs for a long time. The LSTM cell copies its own internal memory

and keeps on accumulating the external signal. This is regulated by structures called gates, which gives LSTM the ability to add or forget information. Gates let the information pass through and are composed of a point-wise multiplication operation and a sigmoid neural network. The sigmoid layer controls how much of the information should be let through by outputting numbers between zero and one.

There are three of these gates in LSTM that control and protect the cell state. The *forget gate* looks at the previous hidden state and the weighted observation/input. The forget gate modifies the cell state by specifying which information to forget by multiplying a position in the matrix by 0 or 1 if the information is to be kept in the cell state. The *input gate* or the *save vector* determines the information that should be stored in long-term memory/cell state. This input gate is also a sigmoid function but since the cell state is a summation of previous cell states, the input gate only adds a number between [0,1] such that the number is not *forgotten*. The *input modulation gate* is another part of the input gate with a tanh activation function of range [-1,1]. This allows the cell to forget memory. The cell state is updated by combining these two gates. The output gate is applied on the cell state to filter the output and decide which parts of the cell state will be sent out.

Using the gating architecture, LSTM solves the vanishing gradient problem as it allows turning 'off' the gate. This can prevent changes to a cell over multiple cycles. Similarly, an 'open' gate does not replace the cell contents at any time, but a weighted average of the previous and the new value is stored.

Bidirectional Long Short Term Memory

A Bidirectional Long Short Term Memory (BLSTM) has two networks, one to access the information in the forward direction and another to access data in the reverse direction. This gives the model the ability to access to the past as well as the future context. BLSTM has been successfully used in image captioning and language translation [88].

2.8. Model Interpretability

Machine learning often involves a trade-off between accuracy and interpretability. A model that outputs correct predictions about the world but offers no insight into the mechanisms involved is not of much use. Especially in bioinformatics, while it is important to have a model with higher accuracy rate, it is equally important to understand what drives those predictions. A model using random forest algorithm is interpretable, considering that the output of random forests is the majority vote by a large number of independent decision trees and each tree is naturally interpretable. On the other hand, deep learning models are black-box and not designed to be interpretable. There has been research on making the deep learning models interpretable, such as LIME (Local interpretable model-agnostic explanation) that approaches the problem by learning an interpretable model in the vicinity (in feature space) of the prediction generated by the ML model [81].

Chapter 3

Methodology

We represent the problem of predicting the RNA binding sites as a binary classification machine learning task. The input is two sets of RNA sequences - positive and negative. The RNA binding protein that is being tested binds to the positive sequences and not to the negative sequences. Given an input RNA sequence of fixed length and an RNA binding protein, we are interested in predicting whether the RBP binds to the RNA sequence or not.

We first use a baseline method based on a Random Forest Classifier and extract sequential and structural features from known PWM profiles. We then use different CNN architectures and evaluate the prediction results on a variety of experimental RBP binding datasets. For the evaluation metric, we report the result as the Area Under Curve (AUC) score for the learned models. In the following sections, we explain the baseline method, feature extraction and classifier pipeline, deep learning architecture, model selection and evaluation stages.

3.1. Data Set

The dataset used for our experiments has been taken from the CLIP-seq data on human genome used in a research study for RBP target sites prediction [84]. This data consists of 19 proteins with one or more experiments for each protein using three protocols (iCLIP, PAR-CLIP, CLIP-seq/HITS-CLIP), totaling to 31 datasets. The dataset was obtained from the servers iCount (http://icount.biolab.si) and DoRiNA [85].

3.2. Baseline Model

We develop a baseline model, generating features using the position weight matrices obtained from RBPDB [76]. The pipeline takes as input an RNA sequence of fixed length of 101 nucleotides and 71 known RBP position weight matrices. The RBP motifs typically have a length of 4 - 12 nucleotides. Taking the motif length as window size, the RNA sequence was scanned and a score was calculated for each position and for each RBP. Each window is a string of length l $(S_1, S_2, ..., S_l)$. For each RBP, we have a position weight matrix P of size lx4, $P[nuc][pos]$. The score is calculated as follows

$$Score_{window} = \sum_{i=1}^{l} P[S_i][i] \tag{3.1}$$

where S_i gives the nucleotide at the position i of the string. The window that closely matches the given RBP motif will have a higher score. If the score exceeded the threshold specified (see next section), the site was considered as a potential binding site. The number of the potential binding sites in a given RNA sequence for a particular RBP was set as the feature value for that RBP. This process was repeated for each of the given RBPs, resulting in a feature vector of 71 values for each RNA sequence.

3.2.1 Selecting the Threshold Value for PWM scores

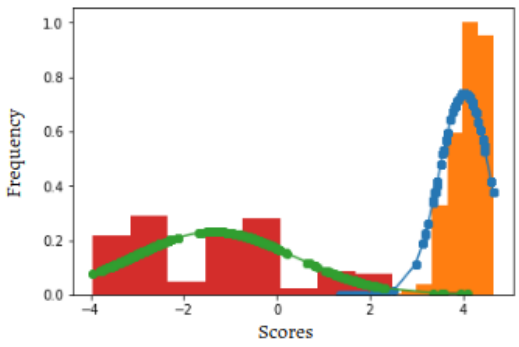

Figure 3.1: Plotting Scores of probable positive (orange) vs. negative (red) sequences for FUS RBP. The green and the blue curve fits a Gaussian over the negative and positive samples respectively.

To select the threshold value for each RBP, we first generated probable sequences of the same length as the RBP motif using Position Frequency Matrices (PFM) obtained from RBPDB. At each position, the nucleotides [A, C, G, U] were selected probabilistically based on their frequencies in the Position Frequency Matrix. This gave a set of probable positive sequences where the given RBP would bind. Similarly, a negative set was constructed for each RBP by randomly selecting the nucleotides for each position (probability of 1/4 for each). The scores for each positive and negative set was calculated using the position weight matrices, and the results were plotted on the graph as given in Figure 3.1. The graph shows the distributions of scores for the positive and negative examples for FUS. The threshold was chosen at a point T where the number of negative sequences with a score lower than T was less than or equal to the number of the positive sequences with a score higher than T.

3.2.2 Random Forest Classifier

Table 3.1: Hyperparameters used in the Random Forest Classifier

Parameter	**Description**	**Value**
n_estimators	number of trees in the forest	500
max_features	number of features to consider when looking for the best split	33% of number of variables
min_samples_split	minimum number of samples required to split an internal node	10% of sample size

The random Forest (RF) classifier from Scikit 3.2.4.3.2. [118] was used to classify the input sequences. The hyperparameters were chosen by multiple iterations of the 5-Fold cross-validation process, each time using different model settings. We then compared all of the models and selected the best one. The best model was used to train the full training set, and then evaluated on the testing set. The hyperparameters used are given in Table 3.1. Each input instance was repre-

sented using 71-feature vectors and the RF algorithm predicts whether the input sequence contains a binding site or not.

3.3. Implementing Deep Learning Architectures

We experimented with several deep learning architectures. In all the architectures, multiple layers are stacked together, and the output of one layer is passed on as the input of the next layer. Three different architectures were constructed, the details of which are given in Table 3.2. We trained the models for 50 epochs. The CNN + BLSTM model converged in 41.08 minutes on average for different experiments with early stopping at epoch 33. The CNN + BLSTM Inception model converged in 25.93 minutes on average with early stopping at epoch 20.

All the architectures shared the basic building structures as shown in the Figure 3.3. Two types of modalities are passed through the input layer. The sequence modality consists of one-hot encoding of the RNA sequences of length 101 as binary vectors using A = 1 0 0 0, G = 0 1 0 0, C = 0 0 1 0 and T = 0 0 0 1, as shown in Figure 3.2.

The structure modality combines the one-hot encoding with the structural probabilities for each position of the RNA sequences as predicted from RNAplfold [65]. RNAplfold predicts the probabilities of bases being in paired/unpaired regions. In order to get the secondary structure predictions, we run RNAplfold on the given sequences. For any given pairs of positions, RNAplfold calculates local pair probabilities for base pairs.

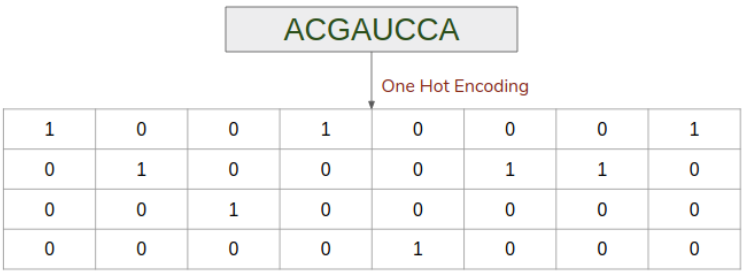

Figure 3.2: One Hot Encoding of an RNA Sequence

In our first model, the CNN has a fully convolutional architecture with a convolution layer, Batch Normalization, an activation layer, max pooling, followed by two dense layers. The CNN + BLSTM model also follows the same architecture but has an extra BLSTM layer. The BLSTM layer processes the output from the convolutional layer, i.e. the feature map, to produce the score from its final time-step.

Table 3.2: The details of deep learning architectures

Model	**Layer 1**	**Layer 2**	**Layer 3**	**Layer 4**	**Layer 5**	**Layer 6**
CNN	conv1d + Batch Normalization (16 filters, kernel_size 10)	ReLU Activation	Max Pooling (size=3)	Dense+ ReLU	Dense+ Sigmoid	-
CNN+ BLSTM	conv1d + Batch Normalization (16 filters, kernel_size 10)	ReLU + Max Pooling (size=3)	BLSTM	Dropout (0.10)	Dense+ ReLU	Dense+ Sigmoid
CNN+ BLSTM + Inception	Three parallel layers (conv1d + Batch Normalization + Max Pooling) (kernel_size = 4,7,11)	Concat parallel layers	BLSTM	Dropout (0.10)	Dense+ ReLU	Dense+ Sigmoid

The convolution layer takes advantage of local correlations in the input such

as sequence motifs and structural contexts and produces a feature map that highlights relevant parts of the input. The LSTM layer can then model complex interactions between different parts of the input. A summary vector is produced that can be used to predict the binding score for the entire sequence. The third model, CNN + BLSTM + Inception architecture was modelled to take advantage of the architecture's ability to specify filters of different sizes in parallel convolution layers.

The function and specific details of the individual layers are explained below.

Convolution Layer

The convolution layer (Conv1D) receives the input signal. It comprises a set of n independent filters of a specific size. Each filter convolves independently with the input signal, and the result is n feature maps of the shape of the input signal. The filters are initialized randomly, and during training, the values of the filters are learned by the network. It is important to randomize the filter values instead of initializing all to 0 or any other fixed value. Random initialization ensures that the filters converge to different local minima in the cost function. We use LSUV initialization [120] as it is very fast and allows for learning of very deep nets. The number of filters and kernel size in a convolution layer is a hyperparameter and after manual tuning, was set at 16 filters of size 10.

Batch Normalization

Along with LSUV initialization, we also introduced Batch normalization (BN) [121] for the layers as it reduces the strong dependence on initialization and solves the problem of covariance shift described below. In a deep neural network architecture, normalization, i.e. shifting inputs to zero mean with a unit variance, is used to make the data comparable across features. As the data flows through the layers, the weights and parameters adjust these values. This can cause the problem of *internal covariate shift*; making the data too big or too small with a small change in initial layers. This is avoided by normalizing the data in each mini-batch, which makes the layers somewhat independent of each other. This helps in speeding up the learning.

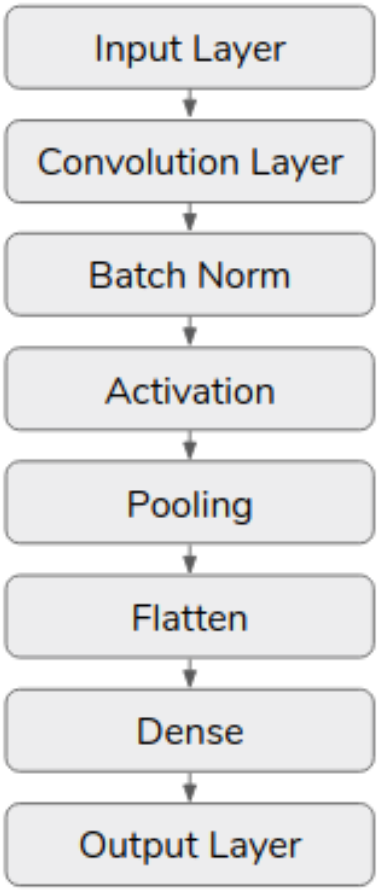

Figure 3.3: Basic building blocks of convolution neural network

Activation Layer

After convolution, the activation layer controls the signal flow from one layer to next. ReLU is the most common activation function used, favoured for its faster training speed. The output of the convolution layer is sparsified, and only the positive values after the convolution operation are given a non-zero gradient value, which helps with the vanishing gradient problem [126].

Max Pooling

The max Pooling (MP) layer reduces the spatial dimension of the output of the convolution layer, which helps in avoiding overfitting. If there are small sequence shifts, pooling captures the dominant component within the region that best summarizes the feature map.

Dense Layer

In a Dense layer, every node is connected to every other node in the previous layer. Dense layer combined with the sigmoid activation function performs classification of the features and gives the probability of the target site being bound to the RNA sequence.

Loss Function and Optimizer

We are using *categorical cross entropy* as our loss function to guide the training process. Using a validation layer, this function gives feedback to the neural network on the predictions made and how far the predictions were from the true value. Using soft-max activation, the cross entropy function can be formulated as

$$Cost = -\sum_{j} b_j log(p_j) \tag{3.2}$$

where b_j is the true output label and p_j is the predicted probability.

For accurate predictions, we need to minimize the calculated error using back-propagation. Using an optimization objective, the weights are modified when propagated back to a previous layer such that the error is minimized. We compared SGD [98], Adam [100] and RMSProp [99] as our optimization algorithm and experimented with a learning rate of 1e-3 and 1e-4 on a variety of experimental RBP datasets. Table 3.3 gives the AUC values obtained and shows that Adam and RMSProp, with learning rate as 1e-3, gave the best results for all architectures.

Table 3.3: Comparing performance by varying learning rates of different optimizers

Optimizer	**Lr = 0.001**	**Lr = 0.0001**
SGD	0.75	0.54
Adam	0.86	0.85
RMSProp	0.86	0.85

Regularization

We also applied dropout, which randomly sets some unit activations as 0 to avoid overfitting the model for training. A dropout rate of 20% was applied, and a performance gain of 2-3% was observed during training. Besides dropout, we also used early stopping combined with batch-normalization to avoid overfitting during the training process.

3.3.1 CNN+ BLSTM model Pipeline

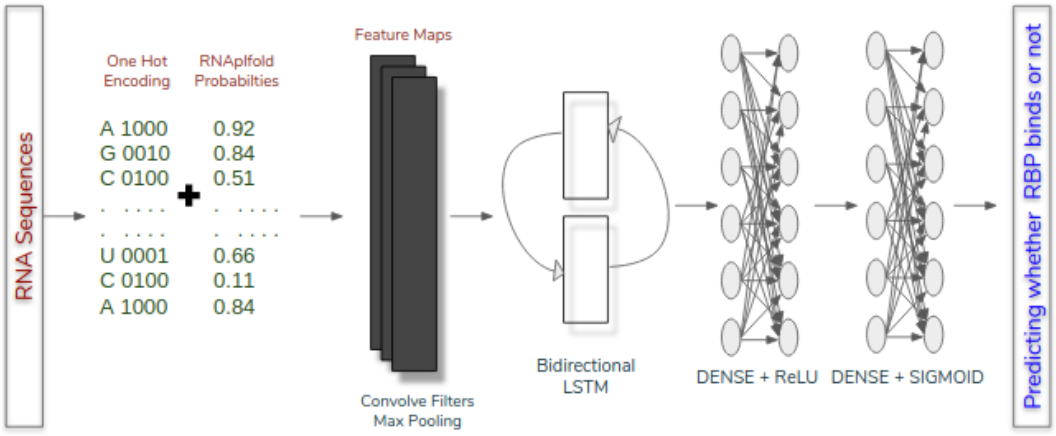

Figure 3.4: CNN + BLSTM Model pipeline.

Figure 3.4 shows the pipeline for CNN + BLSTM architecture. We first encode the RNA sequence into a one-hot matrix and combine it with base pairing probabilities for each nucleotide computed by RNAplfold. The encoded input is

fed into a CNN to output feature maps, followed by a bidirectional LSTM. The bidirectional layer sweeps from opposite directions, and the output of both directions is concatenated for subsequent classification. The BLSTM layer explores long-range dependencies between the sequence and motifs and can model complex interactions between them. The output from BLSTM layer is fed into a Dense layer with ReLU activation followed by another Dense layer with sigmoid activation. The output layer gives us the probability of the RBP binding site prediction. A summary vector is produced that can be used to predict the binding score.

BLSTM Hyperparameters

The number of hidden units in the BLSTM cells were selected randomly from {10, 20, 30}. A higher number of hidden nodes makes the network more powerful. However, it also increases the number of parameters to learn, increasing the time it takes to train the model [88]. We used LSUV initialization [120] for the weights and initialized the biases to a small positive value. The model is trained with mini-batches of 200 inputs using the RMSProp as the optimizer.

3.3.2 CNN+ BLSTM Inception Model

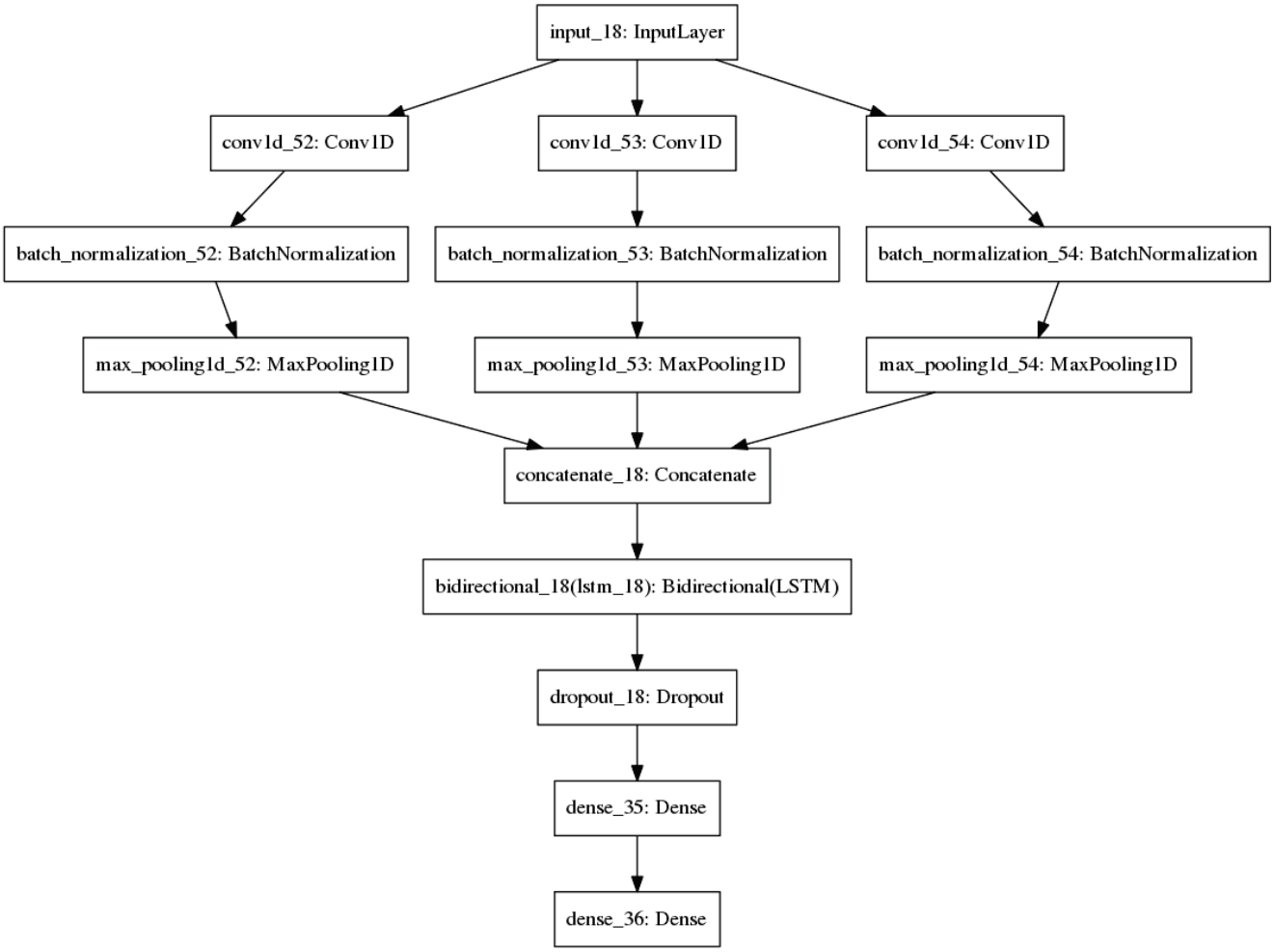

Figure 3.5: CNN + BLSTM - Inception Architecture

The Inception architecture was initially introduced to scale up networks by factorized convolutions and aggressive regularization [130]. It has since been successfully applied to a variety of machine learning problems, particularly in computer vision [127]. The parallel convolutional layers (Figure 3.5) allow us to specify filter lengths of different sizes, corresponding to the fact that the size of RBPs usually ranges from 4-18 nucleotides (from RBPDB PWM dataset). We experimented with a range of parallel convolutional layers to account for different filter sizes. The notion behind using multiple filters is to enable the model to

recognize RBPs of different sizes that the given RBP might interact with. For a given RBP, the model learns appropriate filter weights for different sizes. The layers are concatenated and then fed into the Bidirectional LSTM layer.

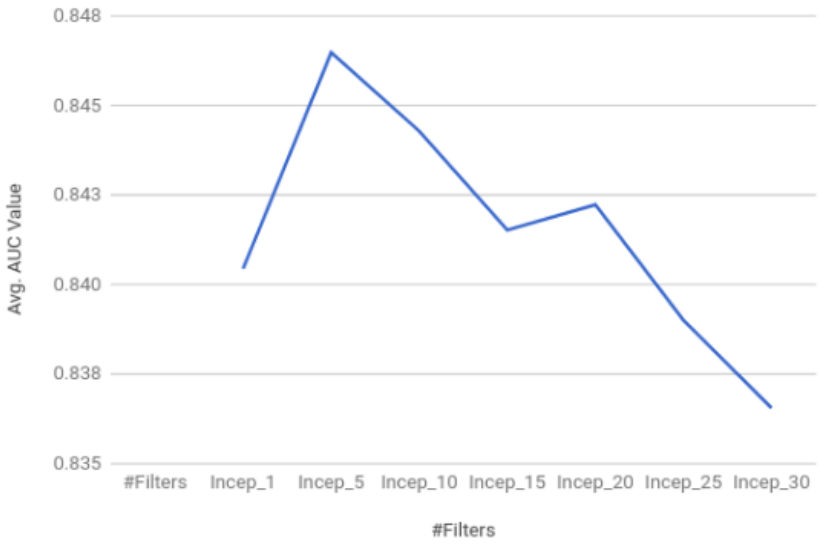

Figure 3.6: Average AUC plotted by varying the number of filters in CNN + BLSTM Inception Model.

The filter sizes for three layers are set as 4, 7 and 11 as the binding motifs are usually of short lengths. We experimented with the number of hidden units for parallel layers of CNN + BLSTM Inception Model and plotted the average AUC across all proteins. Figure 3.6 shows that the model with 5 filters in each parallel layer performed the best and the performance tends to drop after a certain number of filters. However, the difference was not substantial with the range between 0.835 to 0.848.

Implementation Details

The models were implemented in python using Tensorflow 1.5 [124]. The Python code is available at https://github.com/Mishti92/thesis18 .

Chapter 4

Results and Discussion

In this chapter, we discuss about how the data was split and the results obtained from our experiments. We compare the learned models in terms of the AUC score and finally compare our best performing model with state-of-the-art methods. We observed that our model gives better or comparable results compared to other approaches and can learn relevant sequence preferences for the proteins under study.

4.1. Train and Test Data

The data was taken from datasets explained in Section 3.1. For each protein, the data consists of 30000 training examples and 10000 test examples, each sequence of length 101. 80% of the dataset was used for training the model and 20% was held out for validation. The data was split into training and validation randomly.

Table 4.1: Number of positive vs. negative instances

DataSet	**#Positive Examples**	**#Negative Examples**
Train Data	6000	24000
Test Data	2000	8000

The number of negative examples in the dataset is almost twice as much

as the number of positive examples as shown in Table 4.1. Since the data is imbalanced, we will be reporting AUC values for model evaluation after 5 fold-cross validation. We compared our three model architectures and then measured our best performing model against other methods such as Deepbind [71], iDeep [72] and GraphProt [69]. The AUCs reported for Deepbind and Graphprot were obtained from a research study on the same dataset [72] and we evaluated the iDeep tool on the same dataset to get AUC for comparison. We observed that our model outperforms Deepbind and GraphProt and gives comparable results for iDeep for the proteins under study.

4.2. Comparison between Different Model Architectures

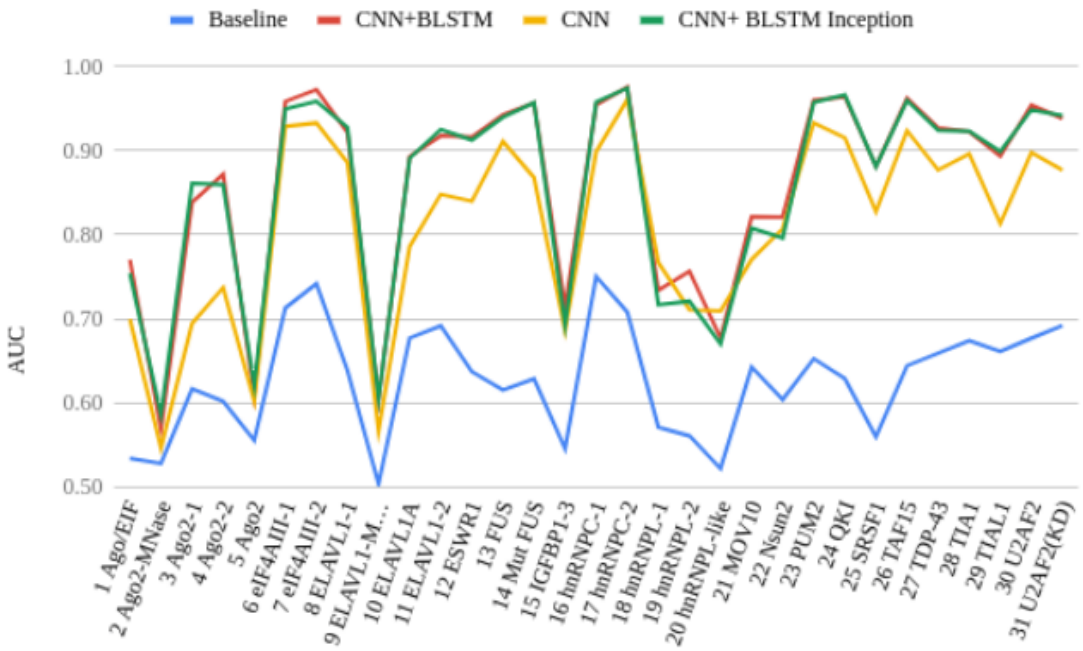

Figure 4.1: Comparing the AUC values of the baseline, CNN, CNN + BLSTM and CNN+BLSTM Inception models.

Figure 4.1 shows the result of comparing the different implemented architectures - Baseline (Random Forest), CNN, CNN + BLSTM and CNN + BLSTM

Inception (explained in Section 3.3). The results show that CNN + BLSTM architecture outperforms our baseline and the CNN architecture for all protein datasets. The baseline method uses random forest classifier with the extracted feature vectors as inputs whereas in CNN based architectures, the model is able to learn high-level features hidden in the original input that improves accuracy. The CNN model tunes the learned parameters automatically after each epoch and learns shared representation of the RNA sequences.

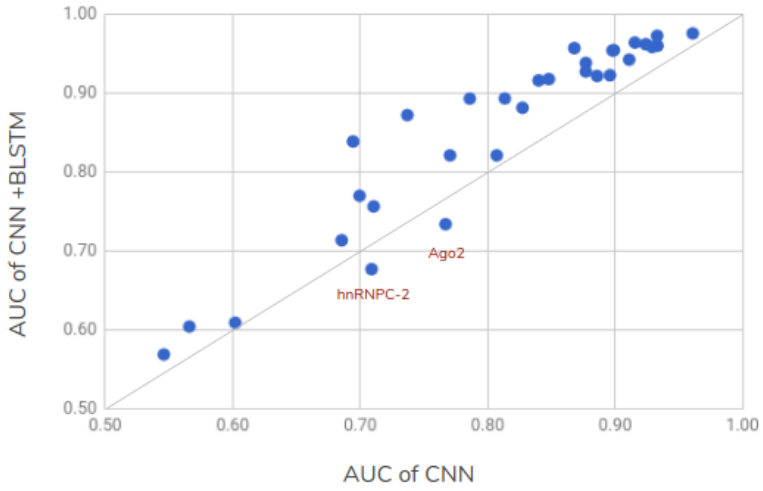

Figure 4.2: Comparing the performance of CNN vs. CNN + BLSTM model. Each dot corresponds to a different RBP dataset.

Figure 4.2 shows the comparison between the CNN and the CNN + BLSTM Model. CNN + BLSTM improves over CNN on almost all proteins. The results indicate that the BLSTM layer is able to capture better motifs for predicting RBP binding sites, which supports that the BLSTM can learn long-term dependencies. On the other hand, it is not clear why CNN + BLSTM model is not learning any new information for hnRNPC-2 and Ago2 protein as their performance doesn't change. The AUC values for CNN+BLSTM and CNN+BLSTM + Inception are similar (Figure 4.3. However, CNN+BLSTM+Inception improves on CNN+BLSTM method for Ago2-MNase and Ago2-1 datasets. Both CNN+BLSTM and CNN+BLSTM+Inception use bidirectional LSTM layer that improves the performance as it learns long-range dependencies between the motifs

and the sequence.

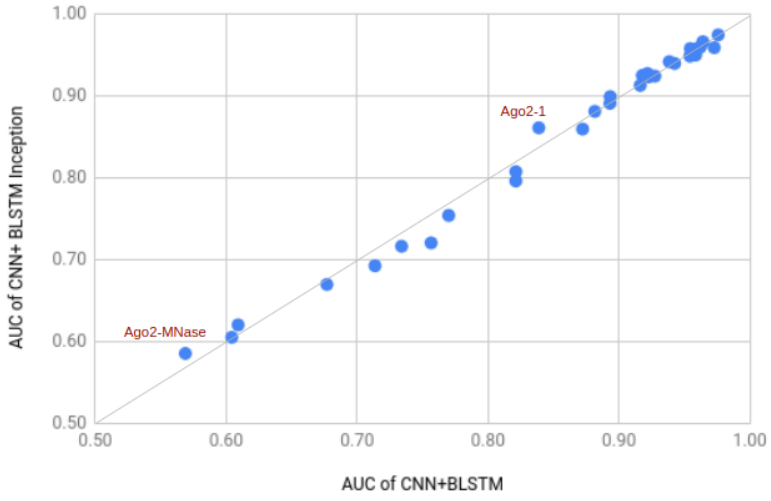

Figure 4.3: Comparing the performance of CNN + BLSTM + Inception vs. CNN + BLSTM.

As can be seen from Figure 4.1, there is a huge variance in the AUC values across the RBPs. For example, the highest AUC value for hnRNPC-2 is 0.98 whereas the AUC for Ago-MNASE for the best performing model is only 0.58. This variability in performance can be seen for all the models. This may be due to the source of dataset for the given RBP, the methodology that was used to identify the binding sequences, e.g. HITS-CLIP, PAR-CLIP, iCLIP, etc. There are cases when a methodology for one RBP performs well but gives poor results for other RBPs. As an example, Ago-2 binding sites extracted using the HITS-CLIP experiment has the average AUC of 0.86 for all models. However, when using the CLIP-seq experiment the average AUC is 0.60. On the other hand, the RBPs extracted using HITS-CLIP performed worse compared to some CLIP-seq experiments. Hence, there doesn't seem to be a correlation between experiments and methodology with their predicted measured performance. The variance in AUC could also be due to the quality of the antibody used. However, not sufficient information is available on how the initial experiments were conducted for the given dataset.

4.3. Binding Motif Preferences

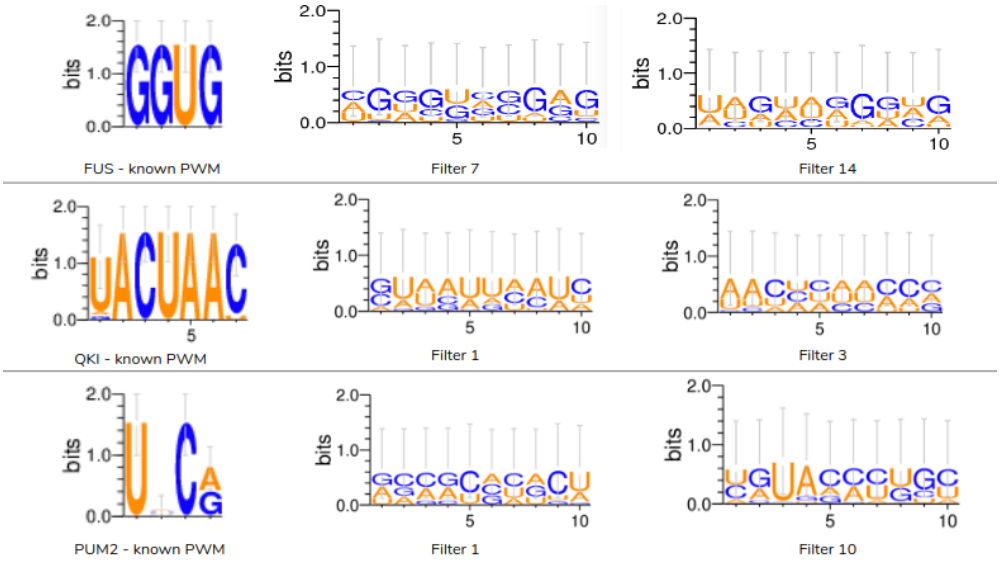

Figure 4.4: The learned filter weights for RBPs FUS, QKI and PUM2 were converted into a PWM and the corresponding matched motif logos were generated using Weblogo tool [132].

To interpret the principles behind the predictions of our model, we explored the learned motifs by investigating the convolved filters of the convolution layer. The filter weights were converted into position weight matrices, and these PWMs were converted into motif logos using a logo generator tool (WebLogo [132]). The generated motifs were matched against the PWMs of known RBPs from RBPDB [76] as shown in the Figure 4.4. We compared three proteins (FUS, QKI and PUM2) with the convolved filters. The PWM of these three RBPs is known which was the basis for comparison. Since a high AUC (> 0.92) was reported for all the three RBP datasets, the filter weights would give us some useful information about the RBPs being predicted. FUS is deterministically known to bind to the "GGUG" motif, and on comparing, we observed that the generated motifs show similar behaviour. The same is true for QKI and PUM2 motifs. These results show that our model can learn binding preferences of the RNA binding proteins and hence, can be used to discover novel binding preferences of the RBPs whose

PWMs are not known.

4.4. Analysis of the Impact of Secondary Structure Data

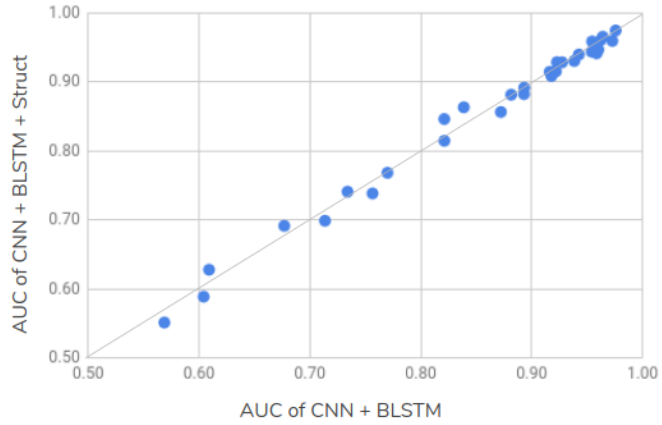

Figure 4.5: Comparing AUC of CNN+BLSTM model trained with (Y axis) and without (X axis) structure input.

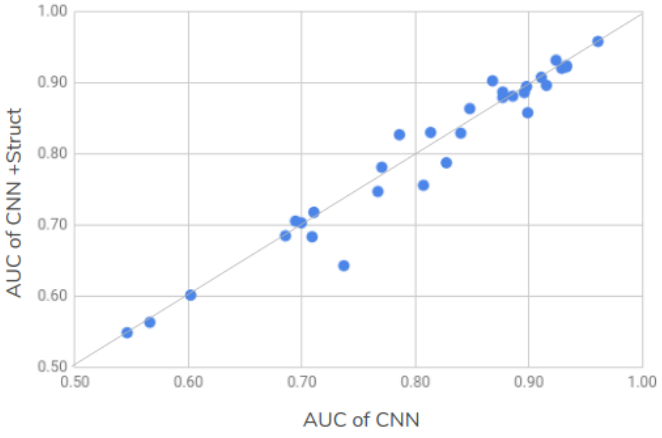

Figure 4.6: Comparing AUC of the CNN model trained with (Y axis) and without (X axis) structure input.

To evaluate the impact of secondary structure in the improvement of predictions we trained our CNN and CNN+BLSTM model with and without secondary structure as input. Both sets of models were trained with the same hyperparameter search space for the sizes of the hidden units. We compared the AUC values on our models using two input types: sequence information and sequence combined with structural information. We observed that for most of the experiments, the sequence input outperforms or gives same results as compared to the combined input modality. These results suggest that the structural input encodes the same information as in the sequential input and the model doesn't learn any new features. However, for some proteins such as Ago2 and ELAVL1A, the structural modality performs better, indicating the structural information for these proteins have more discriminative features.

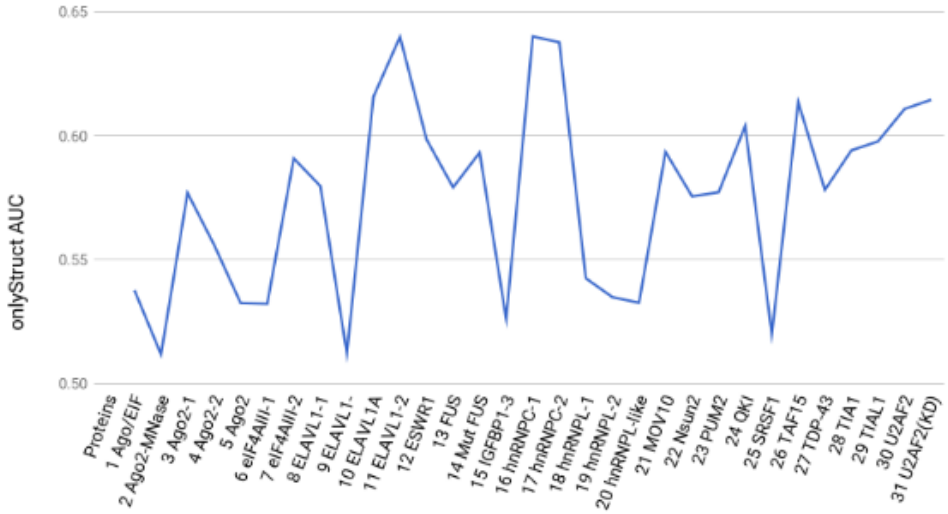

Figure 4.7: AUC values of 31 proteins using only structural probabilities as input.

To better understand the impact of secondary structure on the predictions, we trained the CNN+BLSTM model with only the RNAplfold structural probabilities as input, i.e., without any other information about the sequence itself. The average AUC over 31 proteins was 0.58 with the maximum of 0.64 AUC for hnRNPC-1 and hnRNPC-2 proteins (Figure 4.7). Even though the structural probabilities didn't add much value when trained with sequential input, the AUC values > 0.5 for the structural input alone shows that there is scope to improve our results

with secondary structural features. We performed the Mann-Whitney-Wilcoxon Test (MWWT) [125] to test whether the AUC scores are statistically significant or just random results. For hnRNPC-1 and hnRNPC-2, the average p-value was measured at $5.7 * 10^{-6}$. Based on the p-value, we observe that the results are statistically significant and unlikely to be due to chance.

It can be observed that the model gives higher AUC values for the RBPs hnRNPC-1, hnRNPC-2 and ELAV1-2. These three RBPs are preferentially known to bind to uridine rich sequences [122, 123]. Our assumption is that these poli-U tracts will mostly be unpaired. The structural data involving the said RBPs will have discriminative features encoded within, which could be the reason for their higher AUC values compared to the rest.

4.5. Comparing with state-of-the-art methods

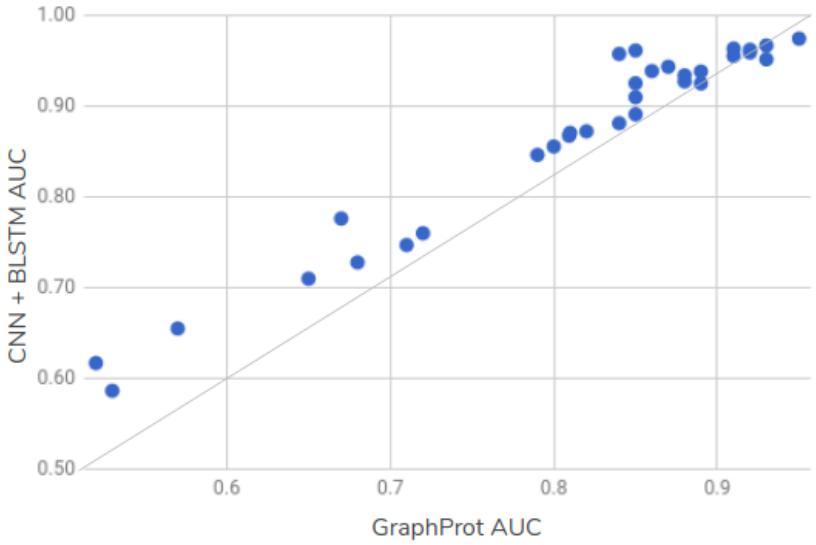

Figure 4.8: Comparing the AUC value of GraphProt and CNN + BLSTM

We compared our best performing model (CNN + BLSTM) with three state-of-art methods in the literature. First, we compared it with GraphProt that incorporates both secondary structural and sequential features using hyper-graphs [69]. Our method outperforms GraphProt on all of the 31 experiments (Figure

4.8). The average AUC over 31 experiments increases from 0.81 for GraphProt to 0.87 for our method. For some experiments such as TAF15 and Ago/EIF, the AUC improves by as much as 15%.

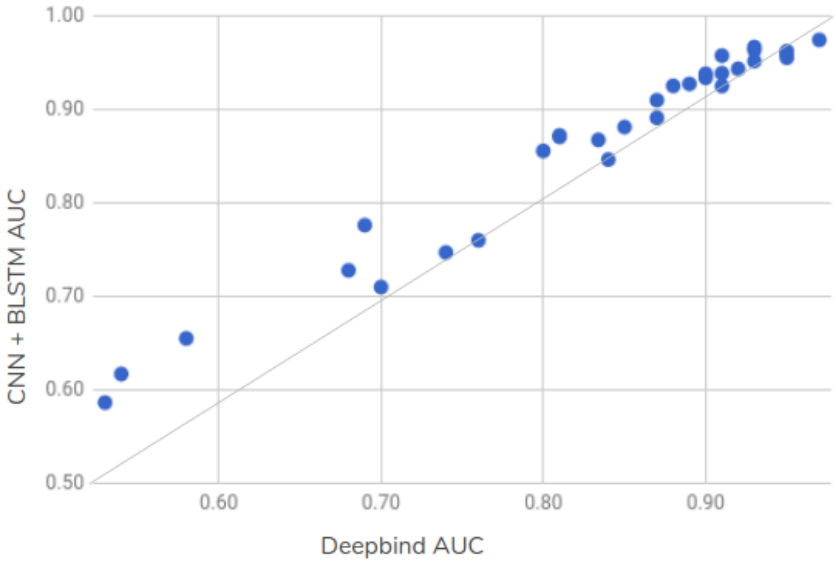

Figure 4.9: Comparing the AUC value of Deepbind and CNN + BLSTM

Our method also performed considerably better than the Deepbind model as shown in Figure 4.9, which achieved the average AUC of 0.83. The AUC values for both models for comparison were taken from the iDeep [72] which used the same data as in this study.

Figure 4.10 shows the comparison between our best performing model and iDeep [72]. The average AUC of iDeep is 0.90 compared to 0.87 for our method. Our model performed better on 5 out of 31 experiments, yielded comparable AUC on 18 experiments and a lower AUC on 8 out of 31 experiments. The high performance of iDeep for some experiments can be attributed to the feature encoding of motif modality using a separate deep belief network integrated with CNN [72].

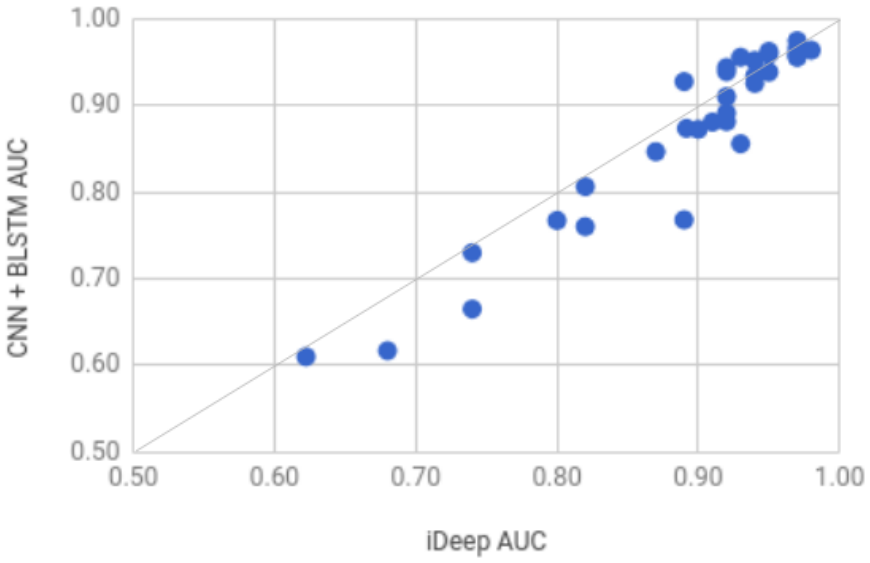

Figure 4.10: Comparing the AUC value of iDeep and CNN + BLSTM

From the results, we observe that the variance in the AUC values for different RBPs is consistent across all models. We assume that the experiments with higher AUC across all models consist of better quality data. The data may contain binding sites with high resolution in the positive examples (possibly due to a better quality antibody that cross-links to target mRNA). Due to this, the models can learn discriminative and distinct features for high-quality datasets, leading to higher AUC values for some experiments compared to others.

Chapter 5

Conclusion and Future Work

RNA binding proteins play an essential role in the post-transcriptional regulation by regulating maturation, degradation, stability and transport of cellular RNAs [2]. Understanding the binding preferences of RBPs also helps us in understanding molecular consequences of RBP mutations in disease, which could lead to better-targeted therapies. Predicting RBP binding intensities can help in the prediction of RNA subcellular localization [128], which is a significant feature for an in-depth understanding of RNA's biological functions. For this reason, prediction of target sites for RBPs has become an important research area in bioinformatics. Experimental methods such as CLIP-seq aim at the genome-wide mapping of RNA binding sites. However, these methods are expensive and time-consuming. The data may contain many false positives due to inherent noise or contamination with non-cross-linked sites, and a large number of binding sites may remain unidentified resulting in a high false-negative rate. These limitations make the task of determining RBP target sites difficult. However, using these high-throughput technologies, a lot of RBP-related genome-wide data is being generated rapidly and stored in public databases such as Protein-RNA interaction database (PRD), which serves as an essential base for computational approaches which can be used to predict RBP-binding sites. In this thesis, we presented a deep learning approach to model RBP binding sites, incorporating both sequential and secondary structure information. Our model captures motifs that align well with the previously reported binding motifs (PWMs) obtained from RBPDB. Our deep learning model contains a bidirectional LSTM layer that captures the

long-term dependencies between the sequence and motifs, which improves the performance of our model. We were able to predict RNA binding preferences with the mean AUC score of 0.87. We compared our results with state-of-the-art methods in the literature and reported improvement in performance over GraphProt [69] and DeepBind [71]. Our method achieved similar but slightly inferior results to iDeep [72]. Compared to traditional black-box machine learning models, we were able to interpret the results producing the human-readable sequence motifs, which can be used to provide valuable information for understanding the biological functions of RNA.

We presented our results on a CLIP-seq data used in a research study for RBP target sites prediction [84]. The training binding sites were derived from a CLIP-seq dataset while the negative sites were taken from genes that are not interacting in any of the 31 experiments. The large variability of reported AUC scores among different RBPs is comparable to the results reported in the aforementioned methods, which indicates a low-quality training dataset rather than a weakness in our method. Thus, in the future, we need to improve the data quality for different RBPs. Knowing the exact experimental steps taken to extract the dataset will help us in evaluating and improving our results better.

We incorporated both sequential and structural features in our model. The AUC scores reported with and without structural features were similar. This shows that the structural modality contains information that has already been captured in the encoded sequences. In future work, one could incorporate structural accessibility information from other tools such as Sfold [58], and encode the RNA structure to 6 elements (stem, multiloop, hairpin loop, an internal loop, bulge and external regions) in order make structural features more discriminative. This would enable the model to learn structural motifs automatically, which will improve the performance. A pipeline could be built separately for the sequential and structural features, and the probabilities at the output layer can be combined.

Our model can further be enhanced to predict the effect of binding sites on mutations. We can mutate the nucleotides of binding sites and test the shift of score on the new site as compared to the experimentally verified sites. It would also be useful to experiment with additional architectures, such as integrating Attention mechanism [131] to focus selectively on the motif subsequences in the

RNA sequence.

An enhanced model that can predict the RNA binding sites more accurately can help in understanding many processes of post-transcriptional regulation such as degradation, stability and transport of cellular RNAs. Human mRNAs on average are 2 kb in length [133] and within mRNAs, different RBPs have different binding preferences. Our model will also help the biologists save time and effort in designing and performing their experiments to detect protein-RNA binding sites by narrowing down candidate binding regions on target RNAs.

Bibliography

[1] Cobb M. 60 years ago, Francis Crick changed the logic of biology. PLoS Biol., 2017, 15(9): e2003243. https://doi.org/10.1371/journal.pbio.2003243

[2] Eliscovich C, Buxbaum AR, Katz Z, Singer R. mRNA on the move: The road to its biological destiny. Journal of Biological Chemistry, 2013.

[3] Horspool D. An overview of the (basic) central dogma of molecular biochemistry with all enzymes labeled. Wikimedia Commons, 2008.

[4] Ray D, Kazan H, et al. A compendium of RNA-binding motifs for decoding gene regulation. Nature, 2013, 499, 172-7. doi: 10.1038/nature12311

[5] Glisovic T, Bachorik JL, Yong J, Dreyfuss G. RNA-binding proteins and post-transcriptional gene regulation FEBS Lett., 2008, 582, 1977-1986

[6] Lukong KE, Chang KW, Khandjian EW, Richard S. RNA-binding proteins in human genetic disease. Trends Genet., 2008, 24(8):416-425

[7] Kyburz A, Friedlein A, Langen H, Keller W. Direct interactions between subunits of CPSF and the U2 snRNP contribute to the coupling of pre-mRNA 3' end processing and splicing Mol. Cell, 23, 2006, 195-205

[8] Buckanovich RJ, Posner, JB, Darnell RB. Nova, the paraneoplastic Ri antigen, is homologous to an RNA-binding protein and is specifically expressed in the developing motor system Neuron, 11, 1993, 657-672

[9] Michlewski G, Sanford JR, Caceres JF. The splicing factor SF2/ASF regulates translation initiation by enhancing phosphorylation of 4E-BP1, Mol Cell, 2008, vol. 30, 179-89.

[10] Vasudevan S, Steitz JA. AU-rich-element-mediated upregulation of translation by FXR1 and Argonaute 2, Cell , 2007, vol. 128, pp. 1105-18.

[11] Chen CY, Shyu AB. AU-rich elements: characterization and importance in mRNA degradation, Trends Biochem Sci, 1995, vol. 20, 465-70.

[12] Barreau C, Paillard L, Osborne HB. AU-rich elements and associated factors: are there unifying principles?, Nucleic Acids Res, 2005, vol. 33, 7138-50.

[13] Sureban SM, Murmu N, Rodriguez P, et al. Functional antagonism between RNA binding proteins HuR and CUGBP2 determines the fate of COX-2 mRNA translation, Gastroenterology ,2007, vol. 132, 1055-65.

[14] De Leeuw F, Zhang T, Wauquier C, et al. The cold-inducible RNA-binding protein migrates from the nucleus to cytoplasmic stress granules by a methylation-dependent mechanism and acts as a translational repressor, Exp Cell Res , 2007, vol. 313, 4130-44.

[15] Duan, R., Sharma, S., Xia, Q., Garber, K. & Jin, P. Towards understanding RNA-mediated neurological disorders. J. Genet. Genomics, 2014, 41, 473-484.

[16] Zhou H, Mangelsdorf M, Liu J, Zhu L, Wu J. RNA-binding proteins in neurological diseases. Science China. Life sciences. 2014, 57. 10.1007/s11427-014-4647-9.

[17] Lukong KE, Fatimy RE. Implications of RNA-binding Proteins for Human Diseases eLS: John Wiley & Sons, Ltd, 2001.

[18] Kim HS, Wilce MCJ, Yoga Y, Pendini NR, Gunzburg MJ, Cowieson NP, Wilson GM, Williams BRG, Gorospe M, Wilce JA. Different modes of interaction by tiar and hur with target RNA and DNA. Nucleic acids research, 2011, 39(3):1117-1130.

[19] Silanes I, Zhan M, Lal A, Yang X, Gorospe M. Identification of a target RNA motif for RNA-binding protein hur. Proceedings of the National Academy of Sciences of the United States of America, 2004, 101(9):2987-2992.

[20] Lukong KE, Chang K, Khandjian EW, Richard S. RNA-binding proteins in human genetic disease. Trends in Genetics, 2008, 24(8):416-425.

[21] Mittal N, Roy N, Babua MM, Jangaa SC. Dissecting the expression dynamics of RNA-binding proteins in posttranscriptional regulatory networks. Proc Nat Sci Acad USA, 2009, 106(48):20300-20305.

[22] Kishore S, Luber S, Zavolan M. Deciphering the role of RNA-binding proteins in the post-transcriptional control of gene expression. Brief Funct Genomic, 2010, 9(5).

[23] Chheda N, Gupta MK. RNA as a Permutation, 2014, eprint arXiv:1403.5477

[24] Draper DE. Themes in RNA-protein recognition. J Mol Biol, 1999, 293(2):255-70.

[25] Weeks KM, Crothers DM. RNA recognition by Tat-derived peptides: interaction in the major groove . Cell, 1991, 66(3):577-588.

[26] Ramos S, Grunert J, Adams DR, Micklem MR, et al. RNA recognition by a Staufen double-stranded RNA-binding domain. Embo J, 2000, 19(5):997-1009.

[27] Sakurambo, Stem loop diagram. Wikimedia Commons, 2015.

[28] Clery A, Blatter M, Allain FH. RNA recognition motifs: boring? Not quite, Curr Opin Struct Biol, 2008, vol. 18, 290-8.

[29] Valverde R, Edwards L, Regan L. Structure and function of KH domains, FEBS J, 2008, vol. 275, 2712-26.

[30] Valverde R, Edwards L, Regan L. Structure and function of KH domains, FEBS J , 2008, vol. 275, 2712-26.

[31] Jenkins HT, Baker-Wilding R, Edwards TA. Structure and RNA binding of the mouse Pumilio-2 Puf domain, J Struct Biol , 2009, vol. 167, 271-6.

[32] Manley JL. SELEX to identify protein binding sites on RNA. Cold Spring Harbor Protoc. 2013 (2),156-163.

[33] Ray D, Kazan H, Chan E, Castillo LP, Chaudhry S, Talukder S, Blencowe BJ, Morris Q, Hughes TR. Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins. Nature biotechnology, 2009, 27(7):667-670.

[34] Keene JD, Komisarow JM, Friedersdorf MB. RIP-Chip: the isolation and identification of mRNAs, microRNAs and protein components of ribonucleoprotein complexes from cell extracts. Nature Prot, 2006, 1(1):302-307.

[35] Brimacombe R, Stiege W, Kyriatsoulis A, Maly P. Intra-RNA and RNA-protein cross-linking techniques in Escherichia coli ribosomes. 1988.

[36] Manley, J. L. SELEX to identify protein-binding sites on RNA. Cold Spring Harb. Protoc. 2013, 156-163.

[37] Bluewhale, Basic Principle of CLIP. Wikimedia Commons. 2014.

[38] Hafner M , Landthaler M, Burger L, Khorshid M, Hausser J, Berninger P, Rothballer A , Ascano M, Jungkamp AC, Munschauer M, Ulrich A, Wardle GS, Dewell S, Zavolan M, Tuschl T. Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. Cell, 2010, 141(1):129-141.

[39] Uhl M, Houwaart T, Corrado G, Wright PR, Backofen R, Computational analysis of CLIP-seq data, Methods, Volumes 118-119, 2017, 60-72, ISSN 1046-2023.

[40] Nostrand E, Pratt G, Shishkin A, Gelboin-Burkhart C, Fang M, Sundararaman B, Blue S, Nguyen T, Surka C, Elkins K, Stanton R, Rigo F, Guttman M, Yeo G. Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP) Nat. Methods, 13 (6), 2016, 508-514.

[41] Fujimori S, Hino K, Saito A, Miyano S, Miyamoto-Sato E. PRD: A Protein-RNA interaction database. Bioinformation. 2012;8:729-730. doi: 10.6026/97320630008729.

[42] Yi Y, Zhao Y, Li C, Zhang L, Huang H, Li Y, Liu L, Hou P, Cui T, Tan P, et al. RAID v2.0: An updated resource of RNA-associated interactions across organisms. Nucleic Acids Res. 2016;45:D115-D118. doi: 10.1093/nar/gkw1052.

[43] Konig, Z Kathi, R Gregor, T Curk, M Kayikci, B Zupan, DJ Turner, NM Luscombe, J Ule. iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution. Nat Struct Mol Biol, 17:909-915, 2010. Methods Enzymol. 164, 287-309.

[44] Foat BC, Houshmandi SS, Olivas WM, Bussemaker HJ. Profiling condition-specific, genome-wide regulation of mRNA stability in yeast. Proc.Natl. Acad. Sci. USA, 2005, 102, 17675-17680.

[45] Bailey TL, Elkan C. The value of prior knowledge in discovering motifs with MEME. Proc Int Conf Intell Syst Mol Biol. 1995;3:21-29

[46] Gerber AP, Herschlag D, Brown PO. Extensive association of functionally and cytotopically related mRNAs with Puf family RNA-binding proteins in yeast. PLoS Biol. 2004;2:E79

[47] Leibovich L, Paz I, Yakhini Z, Mandel-Gutfreund Y: DRIMust: a web server for discovering rank imbalanced motifs using suffix trees. Nucleic Acids Res. 2013, 41: W174-W179. 10.1093/nar/gkt407

[48] Linhart C, Halperin Y, Shamir R. Transcription factor and microRNA motif discovery: The Amadeus platform and a compendium of metazoan target sets. Genome Res. 2008;18:1180-1189

[49] Riordan DP, Herschlag D, Brown PO. Identification of RNA recognition elements in the Saccharomyces cerevisiae transcriptome. Nucleic Acids Res. 2011;39:1501-1509

[50] Li X, Quon G, Lipshitz HD, Morris Q. Predicting in vivo binding sites of RNA-binding proteins using mRNA secondary structure. RNA. 2010;16:1096-1107

[51] Xia T, SantaLucia J, Jr, Burkard ME, Kierzek R, Schroeder SJ, Jiao X, Cox C, Turner DH. Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. Biochemistry. 1998;37:14719-14735.

[52] Mathews DH, Disney MD, Childs JL, Schroeder SJ, Zuker M, Turner DH. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. Proc Natl Acad Sci U S A. 2004;101:7287-7292.

[53] Mandal M, Lee M, Barrick JE, Weinberg Z, Emilsson GM, Ruzzo WL, Breaker RR. A glycine-dependent riboswitch that uses cooperative binding to control gene expression. Science. 2004;306:275-279

[54] Ding Y, Chan CY, Lawrence CE. RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble. RNA. 2005;11:1157-1166

[55] Shapiro, B. A. et al. Bridging the gap in RNA structure prediction. Curr. Opin. Struct. Biol. 2007, 17, 157-165.

[56] Hofacker IL, Fontana W, Stadler PF, Bonhoeffer S, Tacker M, Schuster P: Fast folding and comparison of RNA secondary structures. Monatsh Chem 1994, 125:167-188.

[57] Mathews DH, Disney MD, Childs JL, Schroeder SJ, Zuker M, Turner DH: Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. Proc Natl Acad Sci USA 2004, 101:7287-7292.

[58] Chan CY, Lawrence CE, Ding Y: Structure clustering features on the Sfold web server. Bioinformatics 2005, 21:3926-3928.

[59] Steffen P, Voss B, Rehmsmeier M, Reeder J, Giegerich R: RNAshapes: an integrated RNA analysis package based on abstract shapes. Bioinformatics, 2006, 22:500-503.

[60] Voss B, Giegerich R, Rehmsmeier M: Complete probabilistic analysis of RNA shapes. BMC Biol., 2006, 4:5-27

[61] Li X, Kazan H, Lipshitz HD, Morris QD. Finding the target sites of RNA-binding proteins, Wiley Interdiscip. Rev. RNA, 2014, vol. 5, 111-130.

[62] Frith MC, Li MC, Weng Z. Cluster-Buster: finding dense clusters of motifs in DNA sequences. Nucleic Acids Res. 2003; 31:3666-8.

[63] Ray D, Kazan H, Cook KB, Weirauch MT, Najafabadi HS, Li X, et al. A compendium of RNA-binding motifs for decoding gene regulation. Nature. 2013; 499:172-7. doi:10.1038/nature12311.

[64] Aaron R. Quinlan, Ira M. Hall; BEDTools: a flexible suite of utilities for comparing genomic features, Bioinformatics, Volume 26, Issue 6, 15 March 2010, Pages 841âĂŞ842

[65] Bernhart S, Hofacker I, Stadler P. Local RNA base pairing probabilities in large sequences, Bioinformatics , 2006, vol. 22, 614-615.

[66] Lange SJ, Maticzka D, Mohl M, Gagnon JN, Brown CM, Backofen R: Global or local? Predicting secondary structure and accessibility in mRNAs. Nucleic Acids Res. 2012, 40: 5215-5226. 10.1093/nar/gks181.

[67] Hiller M, Pudimat R, Busch A, Backofen R: Using RNA secondary structures to guide sequence motif finding towards single-stranded regions. Nucleic Acids Res. 2006, 34: e117-10.1093/nar/gkl544.

[68] Kazan H, Ray D, Chan ET, Hughes TR, Morris Q. RNAcontext: a new method for learning the sequence and structure binding preferences of RNA-binding proteins. PLoS Comput Biol. 2010;6:e1000832.

[69] Maticzka D. Lange S.J. Costa F. Backofen R. GraphProt: modeling binding preferences of RNA-binding proteins Genome Biol. 2014.

[70] Zhang S, Zhou J, Hu H, Gong H, Chen L, Cheng C, Zeng J. A deep learning framework for modeling structural features of RNA-binding protein targets. Nucleic Acids Res., 2015, 44, e32. doi: 10.1093/nar/gkv1025.

[71] Alipanahi B. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning, Nature biotechnology, 2015, 33, 831-838.

[72] Pan X, Shen HB. RNA-protein binding motifs mining with a new hybrid deep learning based cross-domain knowledge integration approach. BMC Bioinformatics, 2016, 18, 136.

[73] Pan X, Rijnbeek P, Yan J, Shen HB. Prediction of RNA-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks. 2017, biorxiv 146175.

[74] Kazan H, Ray D, Chan E, Hughes T, Morris Q. RNA- context: A New Method for Learning the Sequence and Structure Binding Preferences of RNA-Binding Proteins. PLoS Comput Biol, 2010, 6(7):1-10.

[75] H.M. Berman, W.K. Olson, D.L. Beveridge, J. Westbrook, A. Gelbin, T. Demeny, S.H. Hsieh, A.R. Srinivasan, B. Schneider The nucleic acid database. A comprehensive relational database of three-dimensional structures of nucleic acids

[76] Cook KB, Kazan H, Zuberi K, Morris Q, Hughes TR. RBPDB: a database of RNA-binding specificities. Nucleic Acids Res. 2011, 39, D301-D308.

[77] Muppirala U. Computational prediction of RNA-protein interaction partners and interfaces. Graduate Theses and Dissertations, 2013, 13610. http://lib.dr.iastate.edu/etd/13610

[78] Lewis BA, Walia RR, Terribilini M, Ferguson J, Zheng C, et al. PRIDB: a Protein-RNA interface database. Nucleic Acids Res., 2011, 39: D277-D282

[79] Yang YC, Di C, Hu B, Zhou M, Liu Y, Song N, Li Y, Umetsu J, Lu ZJ. CLIPdb: A clip-seq database for Protein-RNA interactions. BMC Genom. 2015, 16, 51.

[80] McCaskill J.S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. Biopolymers, 29, 1105âĂŞ1119

[81] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. âĂIJWhy should I trust you?: Explaining the predictions of any classifier.âĂİ Proc. 22nd ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining. ACM, 2016.

[82] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. Nucleic Acids Res. 2000;28:235-242. doi: 10.1093/nar/28.1.235.

[83] ENCODE Project Consortium et al. An integrated encyclopedia of DNA elements in the human genome. Nature, 2012, 489(7414):57-74.

[84] Strazar M, Zitnik M, Zupan B, Ule J, Curk T. Orthogonal matrix factorization enables integrative analysis of multiple RNA binding proteins. Bioinformatics. 2016, 32, 1527-35. doi: 10.1093/bioinformatics/btw003.

[85] Anders G. et al. doRiNA: a database of RNA interactions in post-transcriptional regulation. Nucleic Acids Res., 2012, 40, D180-D186.

[86] Marvin Jens. Dissecting Regulatory Interactions of RNA and Protein: Combining Computation and High-throughput Experiments in Systems Biology. Springer Publishing Company, Incorporated, 2014.

[87] Bottini S, Pratella D, Grandjean V, Repetto E, Trabucchi M. Recent computational developments on CLIP-seq data analysis and microRNA targeting implications, Briefings in Bioinformatics, bbx063, https://doi.org/10.1093/bib/bbx063.

[88] Ray, A., Rajeswar, S., Chaudhury, S.: Text recognition using deep blstm network. In: Proceedings of the International Conference on Advances of Pattern Recognition, 2015.

[89] Guigo R. An Introduction to Position Specific Scoring Matrices. Bioinformatica.upf.edu.

[90] Chen X, Ishwaran H. Random forests for genomic data analysis. Genomics, 2012, 99, 323-329.

[91] Breiman L. Random forests. Machine learning, 2001, 45(1):5-32.

[92] Mueller JP, Massaron L. Machine Learning For Dummies. John Wiley & Sons, 2016.

[93] Claverie J, Audic S. The statistical significance of nucleotide position-weight matrix matches. Computer applications in the biosciences: CABIOS, 1996. 12(5):431-439.

[94] Blackwell A. A Gentle Introduction to Random Forests, Ensembles, and Performance Metrics in a Commercial System. citizennet, 2012.

[95] Staden R. Methods for calculating the probabilities of finding patterns in sequences. Computer applications in the biosciences: CABIOS, 1989, 5(2):89-96.

[96] Zuker M (2003) Mfold web server for nucleic acid folding and hybridization prediction. Nucleic Acids Res 31:3406âĂŞ3415

[97] Murata K, Wolf M. Cryo-electron microscopy for structural analysis of dynamic biological macromolecules. Biochim. Biophys. Acta . 2017; doi:10.1016/j.bbagen.2017.07.020.

[98] Darken, C., Chang, J., & Moody, J. Learning rate schedules for faster stochastic gradient search. Neural Networks for Signal Processing II Proceedings of the 1992 IEEE Workshop, (September), 1-11, 1992.

[99] Tieleman, T. and Hinton, G. Lecture 6.5 - RMSProp, COURSERA: Neural Networks for Machine Learning. Technical report, 2012.

[100] Kingma, D. P., & Ba, J. L. Adam: a Method for Stochastic Optimization. International Conference on Learning Representations, 1-13, 2015.

[101] Greenbaum NL, Ghose R, Nuclear magnetic resonance (NMR) spectroscopy:Structure determination of proteins and nucleic acids. In Encyclopedia of Life Sciences (ELS). John Wiley & Sons, Ltd.:Chichester, 2010.

[102] Baldi P, Brunak S, Chauvin Y, Andersen CA. & Nielsen H. Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics, 2000, 16, 412-424.

[103] Fortmann-Roe S. Understanding the Bias-Variance Tradeoff. scott.fortmann-roe.com, 2012.

[104] Anonymous. Overfitting in Machine Learning: What It Is and How to Prevent It. elitedatascience.com, 2017.

[105] Polamuri S. How the Random Forest Algorithm works in Machine Learning. dataaspirant.com, 2017.

[106] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521(7553):436-444.

[107] Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press, 2016.

[108] Rumelhart D, Hinton G, Williams J, et al. Learning representations by back-propagating errors. Cognitive modeling, 5(3):1, 1988.

[109] Gupta T. Deep Learning: Feedforward Neural Network. towardsdatascience.com, 2017.

[110] Anonymous. Artificial neural network. Wikimedia Commons. 2017. [Online; accessed 04-April-2018].

[111] Clarifai Technology, https://www.clarifai.com/technology. [Online; accessed 06-April-2018].

[112] Krizhevsky A, Sutskever I, Hinton G. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012, 1097-1105.

[113] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, 770-778.

[114] Olah C. Understanding LSTM networks, colah.github.io, 2015.

[115] Gers F, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. Neural Computation, 1999.

[116] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. International Conference on Machine Learning, 2013, 1310-1318.

[117] Kang E. Long Short-Term Memory (LSTM): Concept. Medium. 2017. [Online; accessed 07-April-2018]

[118] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[119] Glorot X, Bengio Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. Proc. Conf. Artificial Intelligence and Statistics, 2010.

[120] Mishkin D, Matas J. All you need is a good init. Proceedings of the International Conference on Learning Representations (ICLR), 2016.

[121] Ioffe, Sergey, Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[122] CienikovÃą Z, Jayne S, Damberger FF, Allain FH-T, Maris C. Evidence for cooperative tandem binding of hnRNP C RRMs in mRNA processing. RNA.;21(11):1931-1942. doi:10.1261/rna.052373.115, 2015.

[123] Ma, W.J., Cheng, S., Campbell, C., Wright, A., Furneaux, H. Cloning and characterization of HuR, a ubiquitously expressed Elav-like protein J. Biol. Chem . 2718144-8151, 1996.

[124] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[125] Michael P Fay and Michael A Proschan. Wilcoxon-mann-whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules. Statistics surveys, 4:1, 2010.

[126] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. Proceedings of the 27th International Conference on Machine Learning, 2010.

[127] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[128] Parton RM, Davidson A, Davis I, Weil TT: Subcellular mRNA localisation at a glance. J Cell Sci 2014, 127:2127-2133.

[129] Hinton G, Srivastava N, Swersky K. Lecture 6a overview of mini-batch gradient descent. Coursera, 2012.

[130] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception architecture for computer vision. arXiv preprint, 1512.00567, 2015. arxiv.org/abs/1512.00567.

[131] Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473, 2014.

[132] Crooks GE, Hon G, Chandonia JM, Brenner SE WebLogo: A sequence logo generator. Genome Research, 2010, 14:1188-1190.

[133] Choi, D., Park, B., Chae, H., Lee, W., Han, K. Predicting protein-binding regions in RNA using nucleotide profiles and compositions. https://doi.org/10.1186/s12918-017-0386-4, 2017.