A Novel RNA Secondary Structure Site Accessibility Prediction Tool using Deep Learning

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
BIOINFORMATICS

Submitted by:

Shubham Mittal

2K18/BIO/10

Under the Supervision of

Dr. YASHA HASIJA

DEPARTMENT OF BIOTECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

JULY, 2020

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Shubham Mittal, Roll No. 2K18/BIO/10 student of M.Tech (Bioinformatics), hereby declare that the project Dissertation titled "A Novel RNA Secondary Structure Site Accessibility Prediction model using Deep Learning" which is submitted by me to the Department of Biotechnology, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any degree, diploma associateship, fellowship or other similar title or recognition.

Place: Delhi **SHUBHAM MITTAL**

Date:

DEPARTMENT OF BIOTECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the project dissertation titled "A Novel RNA Secondary Structure Site Accessibility Prediction Tool using Deep Learning" which is submitted by Shubham Mittal, Roll No. 2K18/BIO/10, Department of Biotechnology, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master in Technology (Bioinformatics), is a record of a project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date:

(Dr. YASHA HASIJA)
SUPERVISOR
Associate Professor
Department of Biotechnology
Delhi Technological University

ABSTRACT

Comprehensive knowledge of base pairing in RNA secondary structure would usher novel insights and develop greater understanding of their role in regulation of cellular processes and in disease. These could then be tackled in a more holistic manner. In pursuit of this objective, probabilistic models, especially those employing machine learning have come to dominate RNA secondary structure prediction, proving better than previous tools which were based upon comparative sequence analysis or folding algorithms employing thermodynamic and stochastic parameter schemes. This study is aimed at developing a machine learning technique better than the previously developed models, which have accelerated the research in RNA secondary structure prediction in the past two decades. The proposed model consists Embedding, CNN and Bidirectional GRU layers which prove effective, when together, for the objective of site accessibility estimation. Specifically, the Gated Recurrent Units (GRUs) are noteworthy since they tackle the problem of vanishing gradient by including the previous and far away time steps for prediction. Data was collected from RNA STRAND database (4666 experimentally determined RNA structures) and Comparative RNA Web (CRW) Site (17032 structures obtained through comparative sequence analysis). From these 4400 structures were curated after cleaning and clustering using CD-Hit. The model was trained, validated, and tested on divisions of this data to give a ROC curve with sensitivity of 0.75 and precision of 0.78, higher than the best compared state-of-the-art RNA structure prediction models, by 11% and 31%, respectively. The ROC values for class 0 with ‘bound’ residues and class 1 with ‘free’ residues were 0.90 and 0.90 respectively, indicating high accuracy in site accessibility prediction. An elaboration on RNA types, functions, and their functional mechanisms in diseases, is intended to provide the reader with the prerequisite knowledge to understand the vitality of unearthing structural information of RNA. Added to this a review of earlier and alternative RNA structure prediction techniques and models is incorporated for a better understanding of the history and scope of RNA structure prediction, through literature.

ACKNOWLEDGEMENT

At the outset I would foremost like to thank my family for their unconditional support and encouragement during this research.

I wish to express my profound gratitude and indebtedness to my guide Dr. Yasha Hasija, Department of Biotechnology, Delhi Technological University, Delhi; for introducing me to this topic and for her inspiring mentorship, constructive criticism and valuable suggestions throughout the project work. If not for her patience, compassion, and expertise the fulfillment of this study would have been a very challenging task. My sincere thanks to PhD Rajkumar Chakraborty for his invaluable help with the code, boundless support, and unwavering optimism.

My profound gratitude also goes to all the research scholars of the Genome Informatics Lab and friends who have extended their help, through their valuable suggestions and helpful discussions, in accomplishing this undertaking.

SHUBHAM MITTAL

(2K18/BIO/10)

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

DNA	Deoxyribonucleic acid
RNA	Ribonucleic acid
A	Adenine
T	Thymine
G	Guanine
C	Cytosine
miRNA	Mirco RNA
mRNA	Messenger RNA
rRNA	Ribosomal RNA
tRNA	Transfer RNA
ncRNA	Non-coding RNA
RBP	RNA Binding Proteins
ANN	Artifitial Neural Networks
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
MALAT1	metastasis-associated lung adenocarcinoma transcript 1
NMR	Nuclear Magnetic Resonance
TRF	tRNA-derived Fragments
ALS	amyotrophic lateral sclerosis
CPU	Central Processing Unit
GPU	Graphics Processing Unit
SCFG	Stochastic Context-Free Grammar

CHAPTER 1 INTRODUCTION

RNA are single stranded nucleic acid molecules – which along with DNA, proteins, lipids and carbohydrates constitute the basic macromolecules that are required by all living organisms. The single stranded nature makes RNA less stable when compared to DNA and therefore more chemically reactive due to which different types of RNA are responsible for carrying out several functions in the cellular mechanisms. Initially RNA structure and functions were limited to the study of mRNA, tRNA and the rRNA. However recent discoveries have determined and established the vitality of the roles of noncoding regulatory RNA, such as, the miRNA, siRNA, piRNA and snRNA. These noncoding RNAs constitute majority of RNA and facilitate mechanisms to inhibit diseases. RNA also exist as enzymes – Ribozymes, and in-conjugation with proteins called RNA binding proteins (RBP). All this is possible due to the highly complex self-folding secondary and tertiary structures formed by RNA. It burgeons the potential of RNA functions and opens avenues for new interactions. Since tertiary structures are very difficult to predict from primary structures due to their dynamic nature and the highly chaotic biological environment in a cell, an attempt is underway to determine the plausible and if possible the best RNA secondary structures from their primary structures and sequences [1]. An RNA strand can self-fold by itself in various ways, forming canonical and nested base pairs, making bulges and helices including stem loops, hairpin loops, internal and external loops, multiloops, pseudoknots, along with non-canonical and non-nested base pairs.

In this endeavor to predict RNA secondary structures several developments have taken place in the past century. Experimental techniques like X-ray crystallography and NMR were used to obtain nucleic acid structures, however since these were too expensive, time taking, and laborious, semi-computational and computational methods came to be developed. Prominent among these are: Mfold (2003), RNAstructure (2010), RNAfold (2011) and RNAshape (2014) – which use experimentally acquired thermodynamic parameters to score structures; and Sankoff method (1985), Knudsen & Hein method (1999) and Alali & Sagot method (2005) – which perform Comparative Sequence Analysis. Additionally, novel experimental techniques like PARS (2010), and those involving DMS (2014) and SHAPE reagents (2008, 2013) were researched.

In this context, it has also been observed over the years that neural networks and deep learning

techniques have been quite successful in generating models for the prediction of biological sequences, functions, and for performing classification. In Artificial Neural Networks (ANNS), CNNs and RNNs prove to be robust in extracting features and generating sequences, which has drawn the attention of the RNA biology academia worldwide. Today, an image classification model without CNNs is difficult to imagine; RNNs on the other hand are highly useful in analyzing sequence data and time series data. Employing such deep neural networks, several tools and models to predict RNA secondary structure have come up over the years. Noteworthy among these being – CONTRAfold (2006), CentroidFold (2009), ContextFold (2011) and SPOT-RNA (2019).

However, among all the methods there remains a performance sealing with respect to accuracy of prediction. A major contributing factor to this is the fact that base pairs, including – lone (unstacked), pseudoknotted (non-nested), and noncanonical (other than A–U, G–C, and G–U) base pairs as well as triplet interactions – are not considered during RNA secondary structure prediction. In fact, separate tools have been developed to predict RNA secondary structures with pseudoknots and others to predict noncanonical base pairs. pknotsRG (2005), Probknot (2010), IPknot (2011), Knotty (2018) and MC-Fold (2008), MC-Fold-DP (2011), and CycleFold (2019) – are some of the profound ones among these. This creates a requirement for a robust method to predict RNA secondary structure with high accuracy, without neglecting its pseudoknots and non-canonical base pairs. For achieving this aim, the foremost requirement is prediction of site accessibility of RNA to estimate the sites that would be available for any binding induced functional roles. The same has been tried to be achieved in this project which employs the Embedded CNN-GRU layers to extract structural features and predicts site accessibility of base pairs with a higher accuracy.

CHAPTER 2 REVIEW OF LITERATURE

2.1 RNA

2.1.1 RNA functions

RNA primary structure is in the form of chains of ribose sugars bound by nucleotide bases and joined by phosphodiester bonds. According to the central dogma of molecular biology RNA primarily function as carriers which enable transfer of genetic information encoded in the DNA into protein molecules. While this is true, recent discoveries point to the existence of RNA largely in the form of non-coding sporadic stretches which play major roles in micro-managing cellular processes, regulating genes and facilitating protein synthesis being a part of ribosomes [2].

Traditionally RNA functions were thought to be confined to three popular RNAs namely – the mRNA: synthesized as a result of transcription, the rRNA: molecules responsible for building the cellular machinery 'ribosomes' along with proteins and the tRNA: adapter molecules, made up of fewer than 100 nucleotides, that facilitate translation of genetic code in mRNA into proteins. In protein synthesis, the genetic code in the messenger RNA (mRNA), produced from DNA as a result of transcription, is in the form of short stretches of 3 nucleotides called 'codons.' Based on the sequence of codons tRNA link corresponding amino acids together to form a polypeptide chain. The entire process is facilitated by Ribosomes and is called 'Translation' [3]. The sequence is popularly known as 'The Central Dogma' – *DNA makes RNA makes Protein*, given by Watson and Crick in 1958 [4]. The mRNA are synthesized in the nucleus and the rRNA are synthesized in the nucleolus and they travel to the cytoplasm for translation; rRNA form the translational unit 'Ribosomes' in the cytoplasm with RNA binding proteins (RBP).

Figure 1: The Central Dogma (Image Source: Koonin, E.V., 2012 [5])

However, in the last three to four decades it has been established that RNA function in higher capacity, with more diverse functions, such as – regulation of DNA replication, facilitating RNA

splicing (post-transcriptional processes) and carrying out biochemical reactions by behaving as catalysts known as 'Ribozymes'. In Eukaryotes they can inhibit, upregulate or downregulate gene expression (Riboswitches) [6] and modify other types of RNA while in prokaryotes they carry out a wide range of process from regulation of bacterial growth to virulence [2]. Furthermore, RNA are known to play crucial roles by serving as biomarkers for several diseases, which emphasizes upon the need to predict RNA secondary structures and understand their functions in a more detailed manner [7].

2.1.2 Types of RNA

Several types of RNA exist in the cellular environment. The mRNA encoded by DNA are referred to as coding RNA (cRNA) – those that code for proteins; besides these the RNA that do not translate to proteins are called noncoding RNA (ncRNA). Noncoding RNA, discovered to be profusely existing, are of two types – the housekeeping RNA, which include the tRNA and rRNA: and the regulatory RNA. Depending upon their sequence length regulatory ncRNA can be divided into long ncRNA (more than 200 nucleotides long) and small ncRNA (less than 200 nucleotides long). The small regulatory ncRNA are of particular functional significance, as it is observed. They are majorly of five kinds – small-interfering RNA (siRNA), microRNA (miRNA), PIWI-interacting RNA (piRNA), small nuclear RNA (snRNA) and small nucleolar RNA (snoRNA).

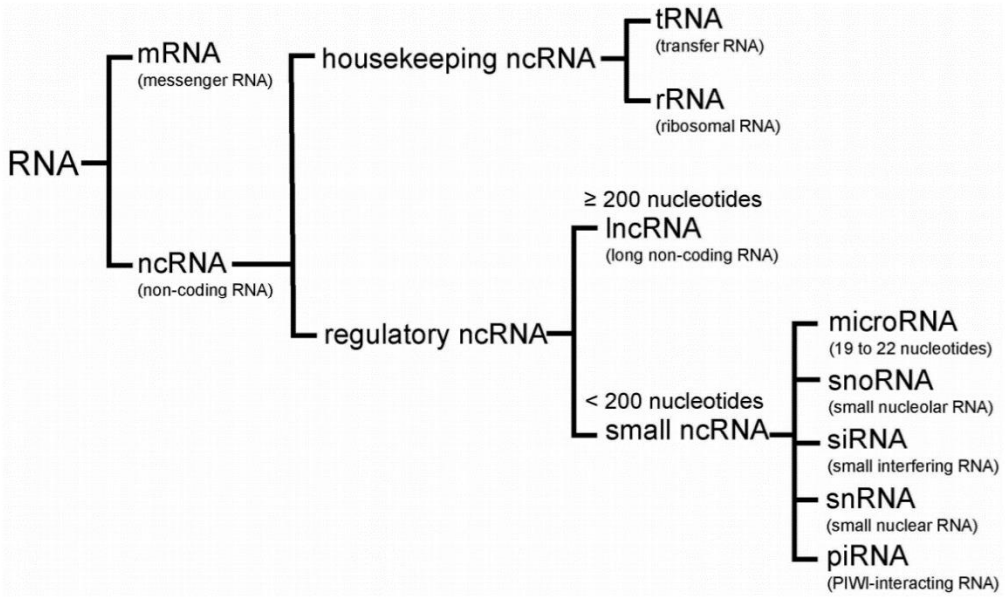


Figure 2: RNA classification (Image Source: Inamura, K., 2017 [8])

The miRNA are endogenous and found in Eukaryotes with a usually 22 nucleotide long sequence;

they attach target mRNA stretches and prevent translation to proteins, thus causing gene silencing. Also, miRNAs are increasingly being used to serve as potential biomarkers for diagnosis of several diseases such as human cancer, while their dysregulation, researches have indicated, can lead to neurological diseases such as Alzhiemer's [9],[10]. Added to miRNAs, the primarily exogenous siRNAs have also been found to cause RNA interference or gene silencing. piRNA are 26-32 nucleotides long RNA which regulate the gene expression of Transposons or 'jumping genes'; they do this by preventing transcription of these genes into germ cells. They are target particular due to specific complementarity and lead to transposon silencing. The snRNA are found in the nucleus and are usually bound to proteins forming complexes called snRNPs or "snurps"; they play a role in gene regulation by splicing pre-mRNA to form mature mRNA. snoRNAs are found in the nucleolus and they consist of RNA families which process rRNA molecules and cause their maturation often by methylation and isomerization of uradine in specific nucleosides [11],[12]. Circular RNA are formed by the fusion of its 3' and 5' loops during alternate splicing; they facilitate protein synthesis but can also bind miRNA, thus preventing miRNAs from carrying out silencing. They also regulate transcription and alternate splicing [13], [14].

2.1.3 RNA diseases

RNA are essential to identifying, understanding, treating and in few cases even supporting human diseases. An example is the case of some types of miRNA that regulate cancer associated genes, in a manner that advances tumour development. Additionally, miRNA dysregulation has been correlated with several neurodegenerative disorders, in particular – the Alzheimer's disease. Besides mRNA other RNA types such as the tRNA are also linked with diseases; for example, the tRNAs have been found to inhibit apoptosis by binding to 'caspases' – the proteins primarily responsible for causing programmed cell death. This facilitates the unhindered proliferation of cells leading to cancer. Also, tRNA-derived Fragments (tRFs) are also being researched as cancer causing agents [10]. Due to the latest RNA sequencing technologies, MALAT1 or "metastasis-associated lung adenocarcinoma transcript 1" – a new groups of RNA transcripts specific to tumours, have been identified; their increased levels point towards growth in quantity and spread of tumour cells because they are found in high amounts in cancerous tissues [15]. Furthermore, few types of RNA have been discovered which are causative in isolating RBPs (RNA-binding proteins) that accumulate in neural tissues of the brain and cause neurological diseases such as ALS (amyotrophic lateral sclerosis) and myotonic dystrophy [16].

2.1.4 RNA Structure

RNA is a single stranded molecule, unlike DNA, its two stranded complementarily bound and more stable cousin. This makes the RNA highly unstable; it possesses high energy and affinity for chemical reactions. Also, the 2′-hydroxyl group on the ribose ring of RNA add to its instability because it enables RNA hydrolysis – where cleavage of phosphodiester bonds linking ribose sugars and phosphate groups takes place, causing the RNA to break. However, RNA are being synthesized in the cell at the same pace as it is being degraded, which holds our cellular makeup and functions. An RNA strand can self-fold by itself in various ways, forming canonical and nested base pairs, making bulges and helices including stem loops, hairpin loops, internal and external loops, multiloops, pseudoknots, along with non-canonical and non-nested base pairs. This self-folding results in three dimensional secondary and tertiary structures, which increase RNA stability and more importantly, it enables RNA to carry out its regulatory functions more suitably. Furthermore, chemical groups such as, methyl, upon binding to RNA three-dimensional structure, further stabilize it. For example, a tRNA sequence with a methyl group bound to its 58th position in molecule make the tRNA more functional and stable when compared to an unstable tRNA molecule devoid of methyl group at the 58th position which eventually gets degraded due to lack of functionality. RNA also forms three dimensional structures by binding to proteins known as ribonucleoproteins (RNPs). In such structures both the RNA and the RBPs function as catalysts [17], [18].

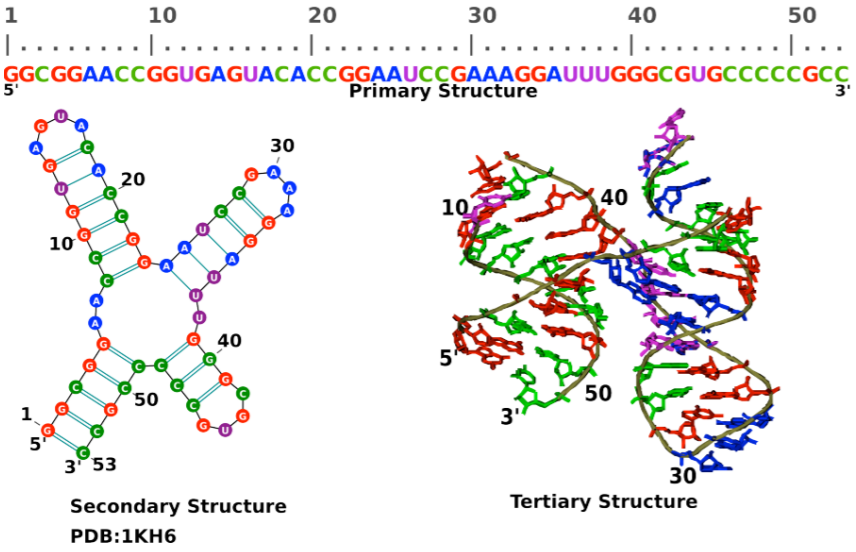


Figure 3: Diagrammatic representation of primary, secondary and tertiary structures of RNA; PDB:1KH6 (Image Source: Kim N. et al., 2013 [19])

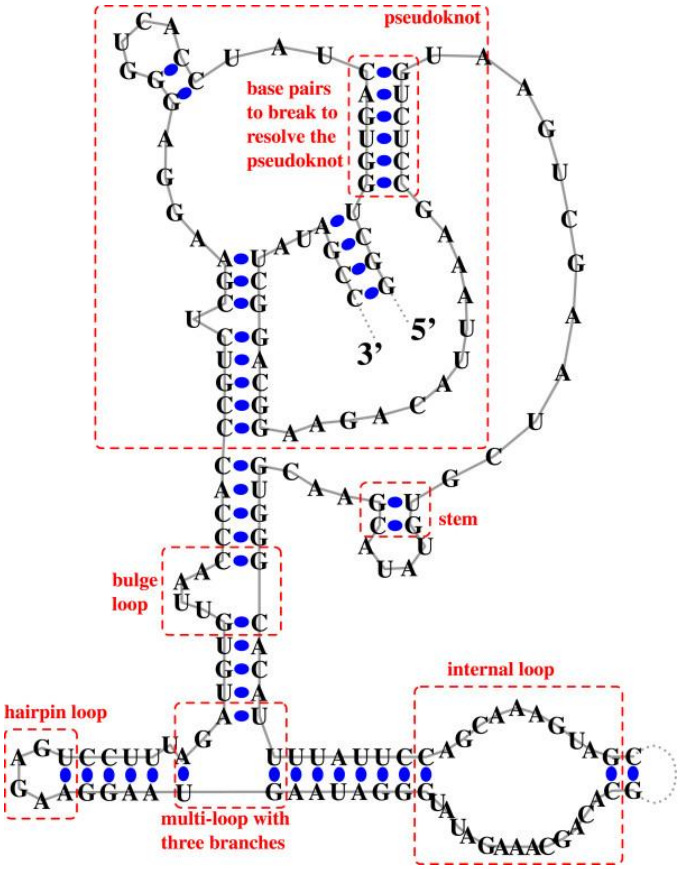


Figure 4: RNA secondary structure with various kinds of loops formed due to self-folding (Image Source: Andronescu M. et al., 2008 [20])

2.2 RNA Secondary Structure Prediction

The discovery of the existence of RNA folding and three-dimensional structures brought with an understanding of RNA functions in greater detail. It became evident that the surfeit of functions of noncoding RNA could be ascribed to the participation of 'free sites' of 3D RNA structures in binding to proteins (RBPs) or other nucleic acid chains (DNA or RNA). Moreover, it is important to note that since tertiary structures are very difficult to predict from primary structures due to their dynamic nature and the highly chaotic biological environment in cells, study of RNA secondary structures from their primary structures and sequences came to be undertaken [1]. To identify the accessible sites on an RNA structure it was important to known how RNAs fold to form structures. This, as was proposed, could be done by obtaining the 3D structures of RNA through experimental techniques, such as, nuclear magnetic resonance (NMR), X-ray crystallography, or cryogenic electron microscopy. However, over time it dawned upon researchers that these techniques are highly time consuming and require adept handling by experts. So much so that only about <0.01%

of the 14 million noncoding RNAs having experimentally determined structures are collected in RNAcentral [21]. Therefore, the need arose to predict RNA secondary structures and thereby identify accessible sites on RNA by non-experimental means. Post this, identifying the nucleic acid bases at these sites would empower us to identify the functions wherein these RNA sites may, or already, play a role – but this would be a subsequent step.

Many other genome-wide structure mapping techniques and computer based techniques have come to be devised over the past several decades for RNA secondary structure prediction. Computational techniques can broadly be divided into two categories – comparative sequence analysis methods and methods involving folding algorithms with scoring schemes of statistical, thermodynamic, or probabilistic nature [22][23]. These are studied in the subsequent sub-sections in the increasing order of relevance.

2.2.1 Physical Methods for Genome Wide RNA Structure 'Mapping'

Apart from the more successful experimental methods to obtain RNA secondary structures, namely, Nuclear Magnetic Resonance (NMR), and the X-ray crystallography, several other experimental attempts were also made to 'map' or 'profile' RNA structure throughout a genome. These are as described below.

2.2.1.1 PARS Technique

PARS, which stands for "parallel analysis of RNA structure", is a technique developed for measurement of an RNA structure at the genome-scale. It employs separate structure-specific endonucleases (enzymes) which cleave double-stranded and single-stranded regions of an RNA to generate a dual library of RNA fragments. Then using deep sequencing, the model analyses the two RNA fragment libraries to identify if the inspected bases were in a double or single-stranded shape, thereby adding to the mapping of entire RNA secondary structure [24]. While the authors used the technique to profile the mRNA secondary structure of *Saccharomyces cerevisiae* (budding yeast), claiming that the coding regions contribute more to the secondary structure than the untranslated regions, they propose that the PARS technique can be used for predicting RNA structures for other species in equal capacity and in diverse conditions.

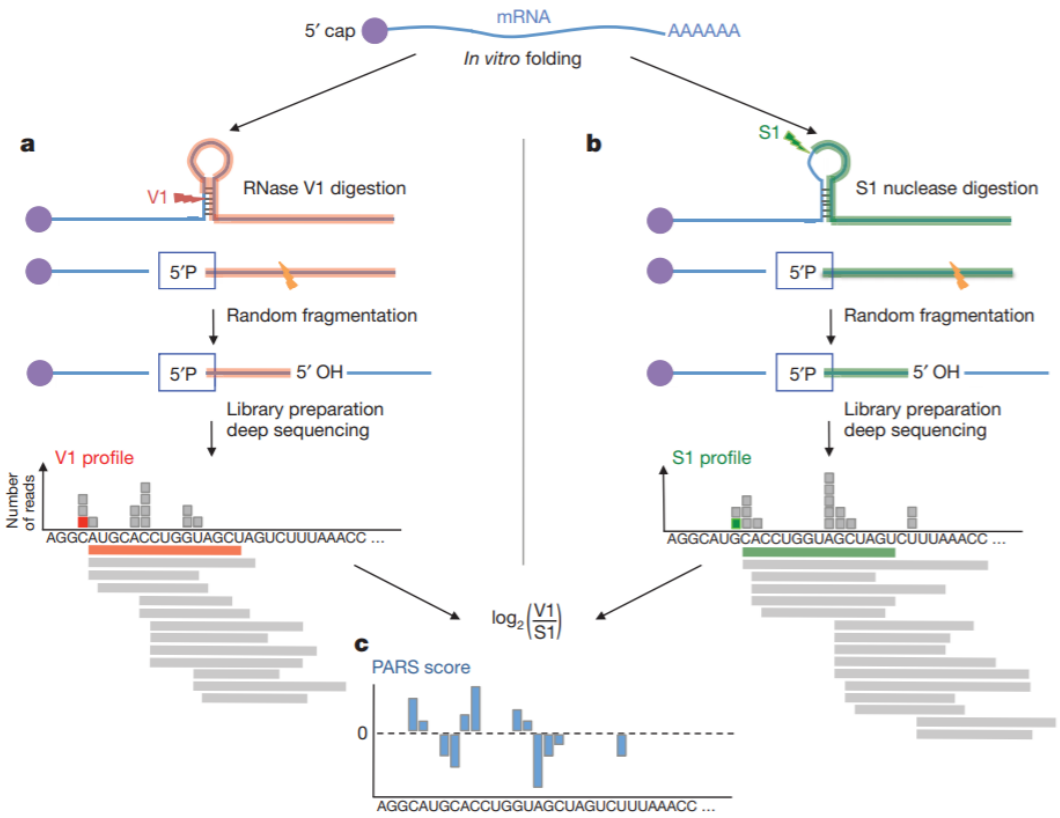


Figure 5: Using PARS technique to acquire RNA structure (Image Source: Kertesz, M., 2010 [24])

However, since the endonucleases are unable to permeate the membrane of a cell, RNA must be extricated out of cells for the technique to be carried out. This causes structural damage in RNA and even induces changes in the RNA natural structure. This remains a major drawback of the PARS technique.

2.2.1.2 DMS Method and the Structure-Seq Model

This method improves upon the problem of endonucleases not being able to enter cell membrane, by using dimethyl sulphate (DMS) that easily enters cells. Going further, the method maps RNA structure, by combining the technique with next-generation sequencing (deep sequencing) to develop a model called 'structure-seq' which enables one to perform *in vivo* quantitative profiling of RNA secondary structure at nucleotide resolution at the genome-wide scale. The method was applied to the genome of *Arabidopsis thaliana* seedlings where initially DMS was used to methylate unpaired adenine and cytosine residues. The RNA sequences were then subjected to reverse transcription, her the methylated As and Cs were not reverse transcribed. This was followed by single stranded DNA ligation, PCR and deep sequencing of the obtained strands, to generate DMS

libraries – positive and control. Upon normalization and subtraction of the 'Reverse Transcription stops' and upon comparing the libraries mapped and unmapped nucleotides of the RNA structure were stated in terms of mRNA and rRNA percentages. The idea behind the method is that constraining even a few nucleotides (As and Cs) – enhances prediction of other regions and this also enables structural ,aping of As and Cs which helps us note the base-pairing status of Us and Gs [25].

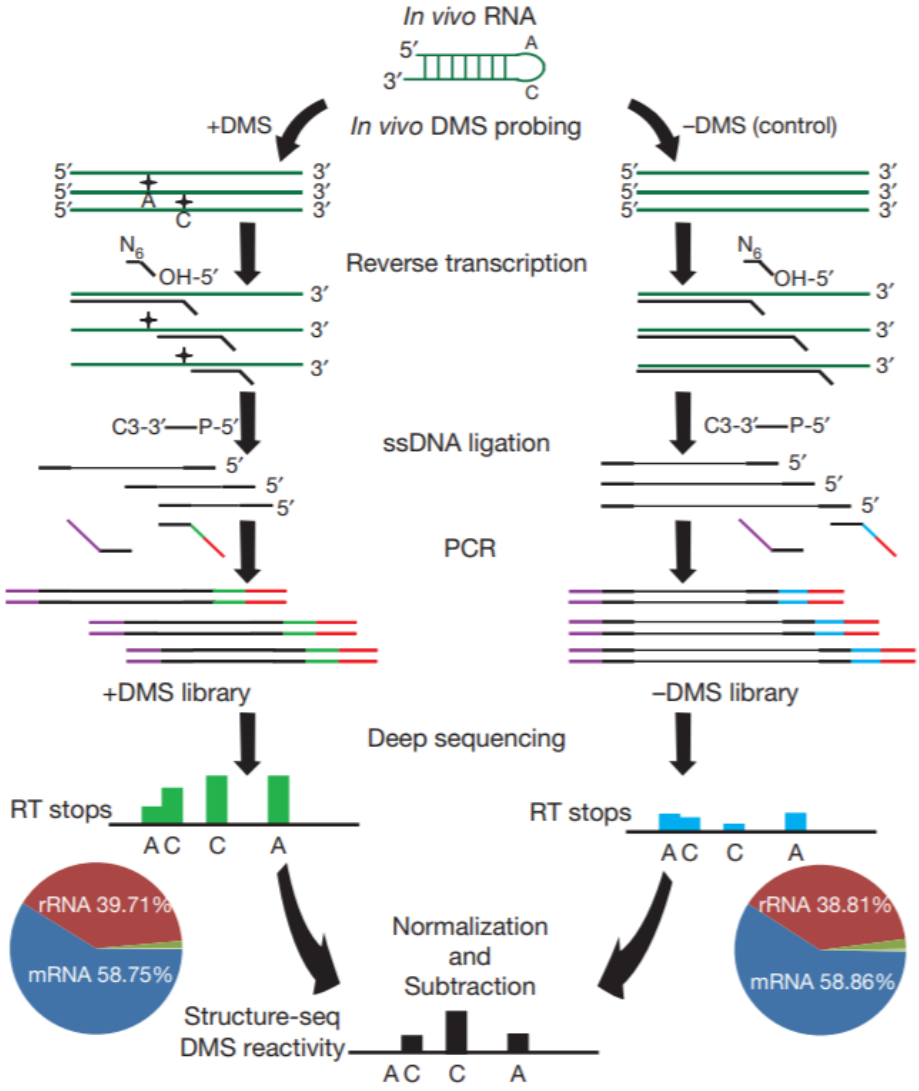


Figure 6: An overview of structure-seq technique (Image Source: Ding, Y., 2014 [25])

While the technique improves RNA structure mapping, it can only identify two paired nucleotides from a molecule of RNA and the estimation of the rest requires simulation using computer algorithms.

2.2.1.3 SHAPE Technique

The current technique was invented in 2008 with the objective of developing a high-throughput, comprehensive and quantitative RNA structure-mapping approach which locates unpaired (flexible) nucleotides within a folded RNA by assessing hundreds of nucleotides at the same time. The study was done by analyzing the genomic RNA structure of HIV-1 virus existing in four biologically significant states, since the function (virulence) depends on shape of RNA folding (structure) and its interactions with proteins [26]. Another study in 2013 employed a 3S (Shotgun Secondary Structure) strategy, that makes use of SHAPE technology, to determine secondary structures of long noncoding RNAs (lncRNA) – the class of RNAs that have emerged as playing a significant role in disease, epigenetics and development [27]. SHAPE (selective 2'-hydroxyl acylation analyzed by primer extension) technology involves acylation of the 2'-hydroxyl of the nucleotide bases A, U, G, C, that exist in an unpaired state; this enables identification of the paired or unpaired nature of bases at a particular position of the RNA structure and also contributes to better understanding flexibility of RNA single strand. However, as a limitation of this method the object of pairing among the paired nucleotides cannot be determined, if it was so then it could have been clearly stated if the bases have self-folded (paired) or are paired to another RNA or protein, which would have remarkably increased its relevance to RNA structure determination.

Drawbacks of Physical Methods for Genome Wide RNA Structure 'Mapping'

Besides the individual limitations of each of the above methods, a larger drawback exists. All the methods propose structure determination through mapping of a single RNA and none of them propose prediction of RNA secondary structure in large quantities. This emphasizes upon the need for computational prediction algorithms to achieve the same.

2.2.2 Thermodynamic Folding Algorithms

First computational algorithms for RNA secondary structure prediction can be traced back to the devising of folding algorithms with thermodynamic scoring schemes. Two major types of mainstream prediction techniques were developed in this regard, as described in the subsequent subsections.

2.2.2.1 Deterministic Dynamic Programming Algorithm

The Dynamic Programming Algorithm till date remains one of the most popular RNA folding

algorithm created by Nussinov et al. in 1978 [28]. It is a simple yet instructive model that aims to predict maximum matching size (MMS) from a given RNA sequence without considering 'crossing' between match loops i.e. the algorithm attempts to predict the maximum number of base pairs forming by means of 'nesting' without considering pseudoknots (crossing) formation. The model claims that the effort to alienate pseudoknots is enforced by base pair maximization which in turn means free energy minimization [28]. As a major improvement over previous methods the algorithm showed promise of being used to predict secondary structure of longer RNA sequences.

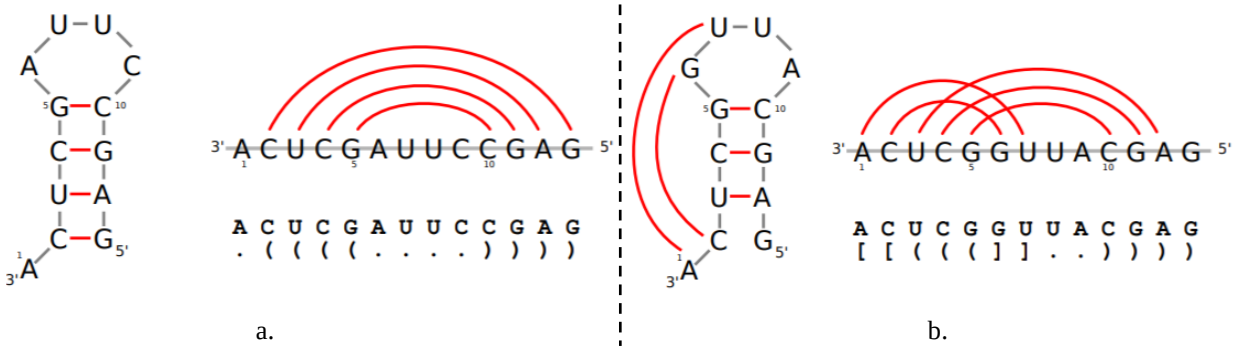


Figure 7: Comparison of RNA folding, one with only nested base pairing (a.) and one with both nested base pairing and crossing i.e. pseudoknots (b.).

However, the algorithm has a low prediction accuracy due to the fact that in reality base pairs do not just form non-crossing 'nested' stem regions; they form multi-loops, internal loops, bulges and pseudoknots – none of which they algorithm encompasses. Furthermore, stacking of base pairs has not been considered and pseudoknots have not been considered. Also, base pair minimization does not give biologically relevant structures, it does not give sub-optimal structures rather focuses on obtaining an optimal structure, which is not natural considering the dynamic nature of RNA tertiary structures.

2.2.2.2 Minimum Free Energy Algorithm

The Minimum Free Energy algorithm (MFE) was developed by Zuker et al. in 1981 [29] and it was modelled after gaining inspiration and insight from the Dynamic Programming Algorithm [28]. The computational technique promised identification of a conformation for a sequence of RNA with the least (minimum) free energy which could be considered as the optimal secondary structure for that RNA. Furthermore, the algorithm made use of efficiency and speed of dynamic programming and combined it with published values of destabilizing and base stacking of RNA to obtain MFE. Moreover, the algorithm incorporated 'additional parameters' that contributed to identification of optimal structure, these being – data on chemical reactivity of RNA and its enzyme susceptibility

[30]. For example, if there is information through enzymatic studies that reflect that under partial hydrolysis condition which phosphodiester bonds are most vulnerable to cleavage, then this information would automatically be built onto the MFE algorithm and together with dynamic programming algorithm RNA optimal structure would be predicted. The efficacy of the model has been shown by demonstrating the folding of a γl heavy chain mRNA fragment that is 496 nucleotide long and has a free energy of -181.4 kcal/mole, a 15% improvement over the minimum free energy found by Rogers et at. [31]. The amalgamation of thermodynamic parameters to perform better prediction has been exemplified by taking two major fragments from 16S ribosomal RNA of *Escherichia coli.* As a positive addition, the model claimed that the energy values are different for different types of folding in an RNA structure (hairpin loop, internal loop, etc) besides being affected by the type of base pairing and the adjacent base pairs. This further appreciates the efficiency of the algorithm.

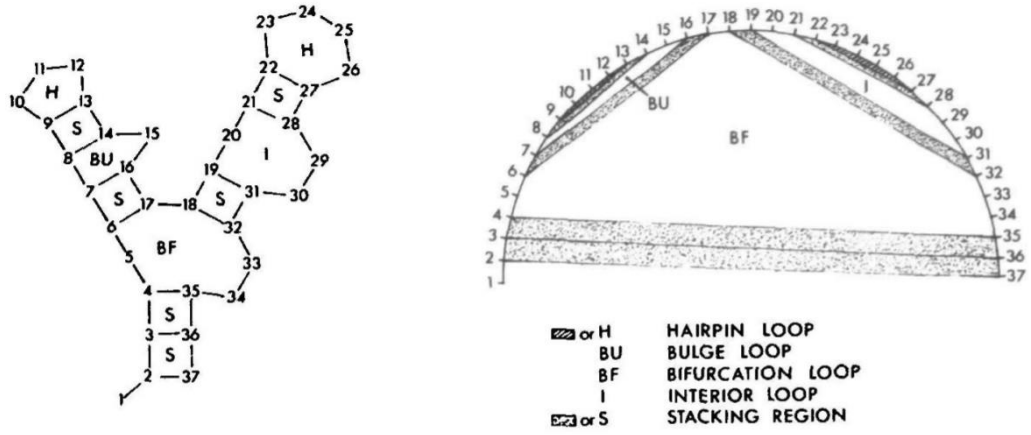


Figure 8: Diagrammatic Representation of an RNA structure including a mathematical graph on the right denoting multiple loop formations. (Image Source: Zuker et al.1981 [29])

Several successful tools came to be developed with the discovery of MFE algorithm.

Mfold – A tool to predict RNA secondary structure using Thermodynamic methods, which is replaced by UNAfold, a tool that replicated and extends mfold by also predicting DNA folding [32],[33],[34].

RNAfold – The tool was proposed as a part of the Vienna RNA Package [35] and was aimed predicting secondary structure of RNA using MFE algorithm along with computations of the equilibrium partition functions and base pairing probabilities. The tool was upgraded with the release of Vienna RNA Package 2.0 [36].

RNAstructure – It is a software package to predict RNA secondary structure, including pseudoknots, and base-pair probabilities based on MFE algorithms and thermodynamic

parameters including 'nearest neighbor parameters' compiled by the Turner group in 2003 [37],[38] and including enzymatic cleavage data, SHAPE data, and chemical modification accessibility.

RNAshape – It came to be understood with time that an MFE structure does is not essentially biologically relevant, so RNAshape tool introduced 'abstract shapes' characteristic which incorporates arrangement of RNA helices while predicting structures [39]. In addition, other works such as maximum expected accuracy (MEA) were also included [40].

Limitations of Thermodynamic Folding Algorithms

Dynamic programming algorithms and subsequent minimum free energy algorithms and further tools improving upon these techniques remarkably increased the overall accuracy of prediction of RNA secondary structure, when compared to earlier methods. However, they mostly focus on obtaining an optimal RNA structure, whereas in natural cellular environment RNA almost never exists in a conformation with minimum free energy. The complex atmosphere in cells induce RNA to stay in suboptimal conformations. Furthermore, Zuker algorithms had a better prediction accuracy for shorter sequences and not for longer RNA sequences.

2.2.3 Comparative Sequence Analysis

Conserved base pairs among sequences that are homologous are determined by comparative sequence analysis. It is commonly believed in biological experiments that among homologous RNA molecules there is a greater conservation of RNA structure than sequence. For example, all tRNA molecules have a clover leaf structure and are homologous despite having less conserved sequences. This is how comparative sequence methods improve accuracy of RNA secondary structure prediction. Some of the popular methods in this study are as follows.

2.2.3.1 Sequence Comparison followed by Structural Prediction

The method predicts secondary structure of RNA by making use of Stochastic Context-Free Grammars (SCFGs) and evolutionary history i.e. homology among sequences. It was proposed by Knudsen and Hein in 1999 [41] which introduces a KH-99 algorithm that first assumes an alignment for all the given sequences and then predicts a single structure based on the alignment. The tool Pfold [42] proposed in 2003 improves upon this model, making it more robust and reducing errors. However, the method does not predict pseudoknots and assumes loop and stem lengths to be

symmetrical. Furthermore, a good alignment is required prior to structure prediction and the time complexity of the model is high.

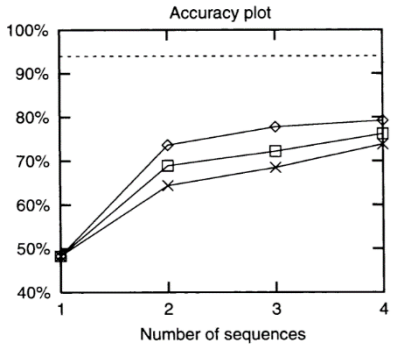


Figure 9: Comparison of accuracy of prediction with and without phylogeny. Diamonds (◊) denote prediction curve with phylogeny, square boxes (□) denote prediction curve without phylogeny and crosses (×) denote prediction curve using CLUSTAL W alignments and phylogeny estimation. Maximum possible prediction accuracy with regards to pseudoknots is represented by the dotted line at 94%. (Image Source: Knudsen & Hein, 1999 [41]).

2.2.3.2 Simultaneous Sequence Comparison and Structural Prediction

This method of RNA secondary structure prediction through comparative sequence analysis proposed by David Sangoff in 1985 concentrated on reducing time rather than memory requirements. It simultaneously performs structural prediction and sequence comparison. However, in its attempt to quicken the process of prediction the model demands excessive computational resources [1]. It's memory requirements escalate to the order of n^{2N}, which serves as a major limitation [43].

2.2.3.3 Structural Prediction followed by Sequence Comparison

The current method proposed by Allali and Sagot in 2005 establishes its model upon the objective to initially predict RNA secondary structure from multiple sequences and then align these structure to identify the most conserved structure to be optimal or to identify common substructures, which could then be used to study unknown structures of potentially similar sequences [44]. The model employs dynamic programming to predict secondary structures from sequences using an algorithm like Mfold [32] and uses tree representation to compare the secondary structures of RNA obtained from the prediction by comparing the distance between them. A novel contribution of the method is an algorithm that introduces 'edge fusion' and 'node fusion' while performing comparative analysis of trees. While this addresses some of the shortcomings of previous classical tree edit operations, the model still does not consider pseudoknots in RNA secondary structure prediction. Additionally, even though the model can predict several candidate structures [1], their validity as

real structures cannot be guaranteed.

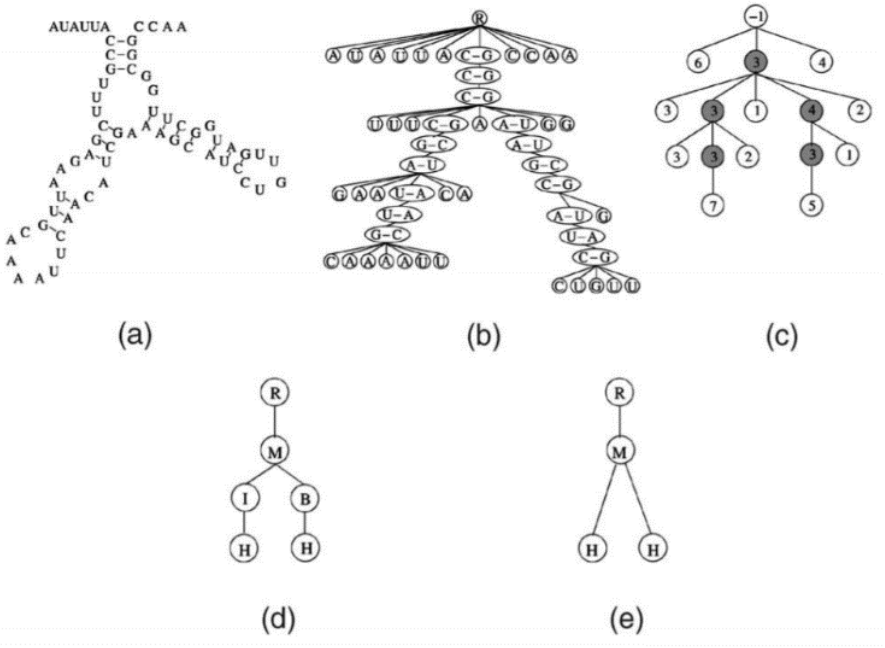


Figure 10: Different tree representation ((b), (c), (d), (e)) of a single RNA (a). (Image Source: Allali et al., 2005 [44])

Drawbacks of Comparative Sequence Analysis Methods

Availability of a large number of such homologous sequences make this method highly successful and precise. However, only about 3000 families are known in Rfam [45], which is insufficient. Furthermore, these methods are not inclusive of pseudoknots and therefore more holistic models are required.

2.2.4 Probabilistic Folding Algorithms

Artificial Intelligence has contributed largely to various fields in the last several decades and with its advent it found applications in RNA secondary structure predictions as well. This was in the form of development of folding algorithms with probabilistic scoring schemes (parameters). In 2003 a unique GPRM model employing a genetic programming method was employed to find common secondary structure elements in a large family with at least 15 members of unaligned RNA sequences [46]. The model did not however find global alignment or single sequence folding. Likewise, in 2006 a neural network method for prediction of RNA folding was proposed with the major advantage of reduced time by improving upon computation complexity of previous stochastic context-free grammar models. Due to the use of heuristic neural network models there was an increased prediction accuracy and the model could be used to predict folding of sequences with

greater than 1000 bases, unlike previous algorithms [47].

Similarly, many other successful probabilistic models were developed generating good results however, they were created on the basis of small sample of data and the accuracy, which though significant, was low for data samples from a single class of RNA. Subsequently, the emergence of deep learning (machine learning) models in the domain of artificial intelligence promised a higher prediction accuracy and a potential for greater inclusion of data for model training to contribute to that prediction. Deep learning models have already showed remarkable performance in predicting protein secondary structure [48], resulting in several advances in the field. While such is the case, it is evident that achieving RNA secondary structure prediction is a more complicated endeavor than protein secondary structure prediction, for the simple reason that while base pairing has to be very specific in RNA (bases can only pair with certain other bases) this is not the case with amino acids, which require no specificity for intermolecular binding. Several successful deep learning models have been developed in the last two decades keeping this complexity in mind. Some of them are described in the following subsections.

2.2.4.1 CONTRAfold

In 2006 researchers from Stanford University proposed CONTRAfold – a novel RNA secondary structure prediction model [49]. This was a landmark model which veered the RNA biology academia into conducting deeper research into probabilistic models for structure prediction. Before CONTRAfold the experimental assays for studying base pairing in RNA sequences were still considered the most reliable [50] even though they proved to be arduous and expensive. After them minimum free energy techniques based on dynamic programming and physics-based empirical methods for thermodynamic parameter determination were the most popular since their performance was unparalleled when compared with any other developed methodologies [29], [32]. However, with increasing advancements in computers and algorithms, SCFGs or 'Stochastic Context-Free Grammars' gained popularity as alternative methodologies for RNA structure determination based on probabilistic algorithms [41], [42]. When energy-based methods employing RNA structural energy based empirical constraints proved insufficient in cases where it was difficult to impose limits on the measurability of certain parameters, the scoring models of SCFGs offered efficient alternatives. The SCFG methods specified grammar rules that encourage joint probability distribution over plausible RNA sequences and structures. The parameters arrived at

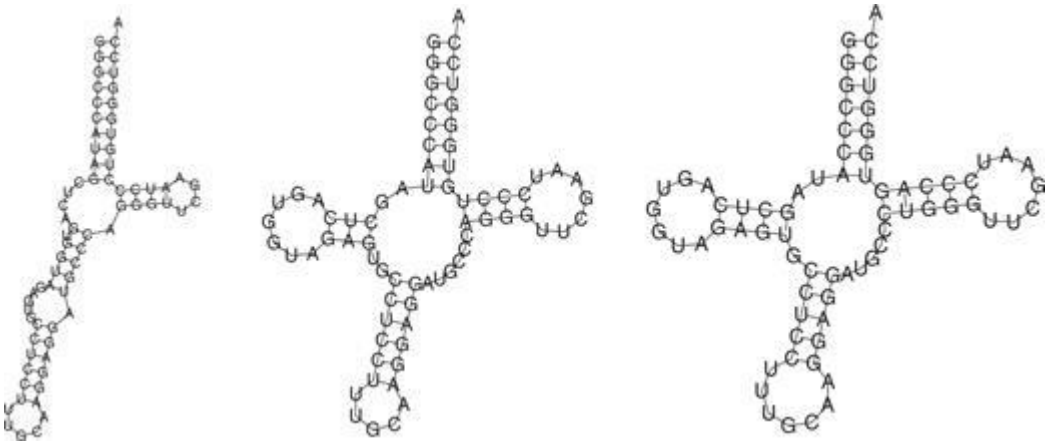
through them were governed by standard mathematical constraints of probability distributions. These parameters could be easily formulated computationally by analyzing clusters of RNA sequences with known RNA structures using SCFGs.

However, even the most complex SCFG models could only show a modest improvement in prediction, which remained much lower than that of empirical thermodynamic models. This propagated a belief among researchers that probabilistic models could never embody the underlying physics that stabilizes RNA structure, in the manner thermodynamic models could. CONTRAfold, however, disapproving this belief, proposed a conditional log-linear model (CLLM) – a flexible probabilistic method to predict RNA secondary structure. CCLMs like SCFGs employ parameter learning and optimization that is driven computationally. But improving over SCFGs, CCLMs generalize by also including complex scoring schemes such as those used in Mfold [32] and other energy-based structure prediction models. They do this through feature rich scoring and discriminative training [49]. This championed CONTRAfold as the pioneer model in the domain of RNA single sequence secondary structure prediction, which used a probabilistic scoring scheme to give the highest accuracy, that was even greater than empirical thermodynamic models. As a limitation, the model does not accommodate for pseudoknots while predicting structure.

2.2.4.2 CENTROIDFOLD

With a focus on predicting noncoding RNA secondary structure, CentroidFold [51] improved upon the previous folding algorithms. The minimum free energy techniques, such as Mfold [32] and RNAfold [52] performed prediction after estimating thermodynamic parameters experimentally. Alternative techniques thereafter were based on probabilistic frameworks which employed stochastic context-free grammars (SCFGs) but did not cater for pseudoknots. These algorithms utilized the popular "Cocke–Younger–Kasami (CYK) algorithm" and calculated the structure with maximum likelihood (ML) and minimum free energy (MFE). However, multiple studies shed light upon the low probabilities of prediction using MFE/ML estimators and lack of optimal prediction of base-pairs. To tackle this limitation several SCFG models considered an ensemble of possible structures as solutions rather than just considering one solution with maximum probability. Alternative estimators facilitated this, for example the Sfold model used centroid estimator [53]–[55] and the CONTRAfold model used minimum expected accuracy (MEA) estimator [49]. These methods maximized the accuracy of prediction. In CentroidFold, the authors proposed a novel

estimator called γ-centroid estimator which promised a higher accuracy of prediction and a superior performance, proven both experimentally and theoretically, when compared to the MEA estimator [51]. The server takes as input both single sequence RNA data (in FASTA or text with plain sequence) or data comprising multiple sequence alignment of RNAs (in CLUSTAL-W format).



a. b. c.

Figure 11: A comparison of a predicted tRNA secondary structure – using RNAfold (a.) and using CentroidFold (b.). (c.) is the reference structure. (Image Source: Sato, K. et al., 2009 [51])

The predicted structures in Figure 11 signify the importance of posterior methods for decoding RNA secondary structures, including γ-centroid estimator and their ability to provide more reliable predictions since several noncoding RNAs do not form reliable secondary structures using MFE algorithms. Notwithstanding its several advantages, the CentroidFold model, like its predecessor does not account for pseudoknots in its predictions.

2.2.4.3 ContextFold

The ContextFold tool was developed in 2011 because of 'rich parameterization' of RNA data [18]. In the race to ever improve the efficacy of prediction models, the authors observed that while methodologies started shifting towards machine learning based probabilistic models, the 'features', the 'parameters' obtained to facilitate structure prediction generally remained consistent in terms of numbers. To increase accuracy the authors aimed at increasing the number of parameters by several fold. What resulted was a model with features for enabling prediction in the order of about 70,000 free parameters and a far more significant improvement in accuracy of prediction; better than the previous best model [56]. Previously the parameters used to be derived from experimental methods [57]–[59] based on the minimum free energy and thermodynamic feature extraction.

However, the increasing availability of both RNA sequential and empirically determined structural data in novel databases [60], the accelerations in RNA folding algorithms [61],[62] and added to this the advancements in machine learning techniques [63],[64] had all collectively made it feasible to improve parameter-estimation remarkably, ensuing greater prediction accuracies. This proved to be appropriate since successful machine learning models are based on large training data [20] which the RNA databases now provided to a significant extent [60]. The structural elements defined by the ContextFold tool are based on those of the Turner99 model [65] for enabling effective interfacing. However, unlike any previous scoring models more structural elements and longer sequences comprise the structure of this model.

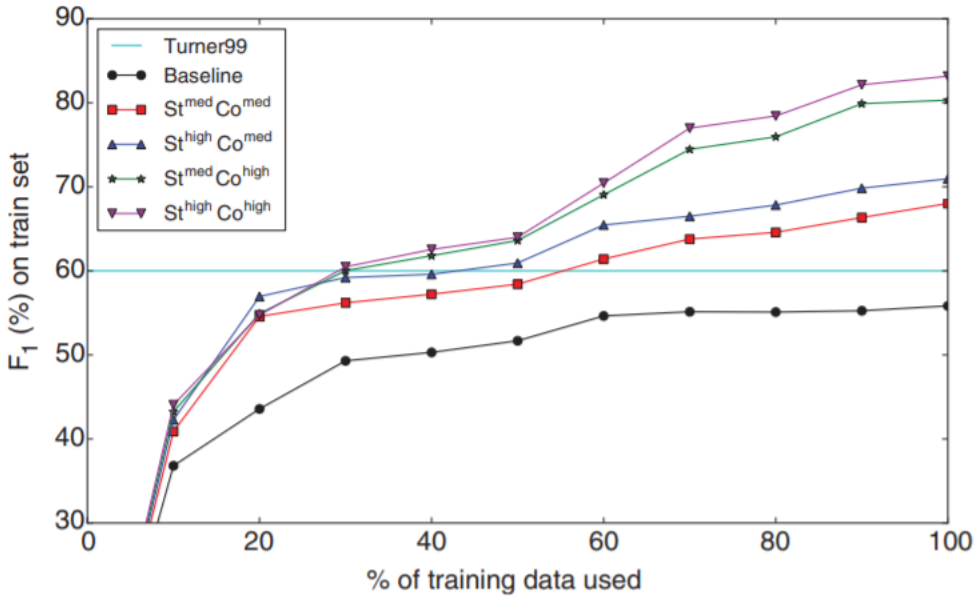


Figure 12: A demonstration of increase in validation set accuracies with increase in training set size (Image Source: Zakov S. et al., 2011 [18])

Moreover, while earlier models assigned only a single score to each element, ContextFold assigned score elements as a summation of scores of various sub features encompassing the wide array of structural and sequential data. This made the model more detailed and robust with a higher accuracy for prediction. ContextFold boasted an ensemble of 70,000 parameters, which is a remarkable improvement over parameters in previous models by manifold. The model claimed an error reduction rate of about 50% compared to previously known best models, a remarkable feat [18]. Like CONTRAfold and CentroidFold models however, ContextFold accounted only for pseudoknot-free RNA folding.

2.2.5 Additional Methods

2.2.5.1 Developments for RNA tertiary structure prediction

In any of models described in the above methods, the accuracy of prediction could not cross 80%. This ceiling in performance existed largely due to the exclusion of base pair interactions that result in the formation of tertiary structures in RNA [22], [66], [67]. Such interactions can be – unstacked base pairs, pseudoknots (non-nested base pairs) - especially kissing hairpins, noncanonical base pairs (that do not follow base-pairing proposed by Watson-Crick i.e. G-C, A-U and G-U) and triplet interfaces [13], [21]. In context of this realization some noteworthy RNA secondary structure web-tools were developed encompassing these interactions.

pknotsRG – It employs an algorithm for prediction of RNA secondary structures including pseudoknots under the minimum free energy (MFE) model in $O(N^4)$ time, where N is the length of the sequence. The class of pseudoknots considered by the algorithm are 'simple recursive' which are restricted by the three canonization rules [68].

Probknot – A tool for predicting RNA secondary structure with pseudoknots with a reduced time of $O(N^2)$, where N is the length of the sequence. Probknot assembles MEA structures from calculated probabilities of base pairing [69].

IPknot – Another RNA secondary structure prediction tool including pseudoknots. It improves upon the limitation of analyzing only certain types of pseudoknots by studying a wider class of pseudoknots with decreased time. It further shifts from an MEA based algorithm to a heuristic integer programming algorithm to refine base-pairing probabilities. IPknot can also predict consensus secondary structure from a multiple sequence alignment input and it gives a better prediction accuracy [70].

Knotty – It improves upon the preceding models on predicting RNA secondary structures including pseudoknots in terms of space and time consumption. Previously the authors proposed a CCJ 1.0 algorithm that essentially involved recognizing and predicting TGB (three-groups of bands) pseudoknot structures [71]. In Knotty (CCJ 2.0) the authors claim improved space complexity due to the technique of 'sparsification'. The model outperforms CCJ 1.0 [71] and Pknots [72] in terms of run-time. Moreover, it employs 'HotKnots DP09' [73], a state-of-the-art energy model, for a superior

accuracy of prediction [74].

MC-Fold and MC-Sym Pipeline – The authors propose a model for predicting non-canonical base pairings (i.e. other than A-U, G-C and the occasional G-U wobble) in RNA secondary structures, in an effort to understand the tertiary structures of RNA. The model considers all base-pairing possibilities as scoring functions for RNA folding. MC-Fold and MC-Sym are two algorithms based after cyclic motifs of nucleotides, which upon considering these scoring functions showed promising folding prediction results from sequences [75].

MC-Fold-DP – It is a theoretical model that captures extended structural motifs of RNA that share nucleotides and RNA pair families, thereby predicting non-canonical base pairing in RNA secondary structures. The model adds dynamic programming to add sparse corrections of data to the earlier MC-Fold method [75] for better performance and proposes a number of programs for optimizing parameters and predicting structures [76].

CycleFold – A knowledge-based model is the thrust of this method which predicts non-canonical base pairing by scoring nucleotide cyclic motifs (NCMs). A partition function algorithm is proposed which estimates, for both non-canonical and canonical base pairs, the base-pairing probabilities. A further improvement of these base-pairing probabilities estimations is facilitated by the previously published TurboFold algorithm [77]. Complementing this the previous knowledge of canonical secondary structures result in greater prediction accuracy of non-canonical base-pairs, resulting in a more precise prediction of complete tertiary structures [78].

2.2.5.2 Hybrid Methods

While research aimed towards predicting RNA structure diverged into different classes depending on type of algorithms used, some researchers attempted at applying a hybrid approach towards solving the problem. These studies are far more recent and are increasing, propagating ideas to develop better techniques and algorithms. One such example is the **CDP-Fold** algorithm [1] that employs convolutional neural network and dynamic programming along with sequence alignment. While the method results in a higher accuracy compared to previous singular algorithms, they still hold some drawbacks. For example, the problem of G-U swing pairs (or wobble) is not well-explained and integrated in the algorithm. The results yet do not give a satisfactory secondary structure, which the authors feel can be improved through better optimization. Also, the model considers very few numbers of longer sequences in input data which must be increased; and a better

prediction of pseudoknots must be incorporated. Another technique, called **SPOT-RNA** [23] was proposed, aimed at enhancing the modeling of RNA structure, their sequence alignments and their functional annotations. The model employs techniques of deep learning, such as convolutional neural networks, two-dimensional Bidirectional Long Short-Term Memory (2D-BLSTM) and combines it with transfer learning on data from bpRNA – a large repository of about 10,000 RNA sequences. The algorithm proposed, claims to include all sorts of RNA structural interactions including those involved in forming a tertiary structure. These are – unstacked (lone) base-pairs, pseudoknots, non-canonical base-pairs, and triplet interactions. A limitation is that the data used as input does not have many large RNA sequences, however the computational time taken by the model is favorably decent for genome-scale studies.

2.2.5.3 RNA Inverse Folding

In pursuit of attaining structure from sequence to understand function of RNA, researchers tried another approach on the side: designing desired RNA sequences depending on the target secondary structure required for specific functions. Several top-of-the-line algorithms have been developed in this approach, such as, the local search algorithms, constraint programming and structure algorithm, downhill simplex algorithm, multi-objective genetic algorithm, etc. Some of the popular tools for inverse folding are **RNAinverse** [52], **EteRNABot** [79], **Frnakestein** [80], **RNAiFold** [81], **IncaRNAtion** [82] and **MODENA** [83]. All these, though moderately satisfactory in complicated cases, are found to be decent overall for RNA inverse folding [84].

Table 1: Comparison of Popular RNA Secondary Structure Prediction Methods in last 20 years.

Name	Algorithm	Machine learning used	Pseudoknot Prediction	Reference
PARS (2010)	Comparative Sequence Analysis	No	No	[24]
DMS (2014)	Comparative Sequence Analysis	No	No	[25]
SHAPE (2008)	Comparative Sequence Analysis	No	No	[26]
RNAshapes (2014)	Thermodynamic	No	No	[40],[39]
RNAstructure (2010)	Thermodynamic	No	Yes	[38]
RNAfold (2011)	Thermodynamic	No	No	[36]
CONTRAfold (2006)	Probabilistic	Yes	No	[49]
CentroidFold (2009)	Probabilistic	Yes	No	[51]
ContextFold (2011)	Probabilistic	Yes	No	[18]
SPOT-RNA (2019)	Probabilistic	Yes	Yes	[23]
Knotty (2018)	CCJ 2.0	No	Yes	[74]
IPknot (2011)	MEA and Heuristic Integer Programming	No	Yes	[70]
MC-Fold and MC-Sym Pipeline (2008)	MC-Fold and MC-Sym	No	Yes	[75]
CDP-Fold (2019)	Probabilistic and Dynamic Programming	Yes	Yes	[1]

2.3 Neural Networks

2.3.1 Machine Learning General

Machine learning can be defined as a branch of artificial intelligence in which systems, trained on known data, facilitate prediction of patterns in unknown data due to generation of mathematical models. Techniques of machine learning are approaches to study data and gain operational relationships from them without defining them beforehand [85], [86]. Coming to computational biology, the appeal of machine learning is its ability to deduce prescient models without requiring any solid knowledge concerning the causal processes, that are much of the time obscure or inadequately characterized. For example, most precise forecasts of expression of gene levels are presently produced by a wide range of epigenetic characters utilizing random forests [87] or sparse linear models [88]; the mechanism behind obtaining transcript levels from the chosen features, however, remains a topic of active study. Machine learning methods remain integral for predictions in the fields of biotechnology. Be it metabolomics [89], genomics [90], proteomics [91], or assessing drug sensitivity [92].

Four stages comprise the general machine learning methodology which describes the above applications. These are: cleaning of data and its preprocessing, mining and determination of attributes (features), ascertaining fitting of model and finally assessment, which gives results (Figure 13). As per convention the sample of one data is demarcated as *info x*, which is usually more than one number, and its output value is labelled as *y (output)*, which is usually a solitary number [93].

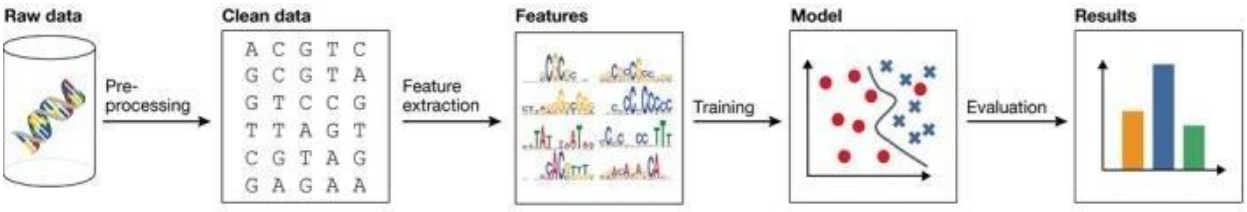


Figure 13: The standard machine learning model can be divided into four steps: pre-processing of data, feature extraction, model learning and evaluation of model (Image Source: Angermueller C, et al., 2016 [93]).

Machine learning models are popularly divided into supervised and unsupervised. A **supervised model** attempts at obtaining relationships (functions) like $F(x) = y$, from recorded data pairs used for training, such as (x_1, y_1) and (x_2, y_2) and so on. For example, in biology this can be classically applied to study the effect of a compound (drug) on cancer cells [92]. Variants in sequence in

somatic cells of the cell line would be encapsulated by *x* (input features), along with the drug's concentration and chemical composition, while the effect on cell lines (viability) would be represented by the label *y* (output). These labels together can train a random forest classifier or an SVM (support vector machine) or other similar methods to determine the relationship *F*. In the future, the trained function *F(x')* can be used to determine the output label *y'* for a new cell line with input data *x'*. This occurs, even if the series of steps bringing about the relationship identified by the function *F*, between sequence mutation and cell line viability are not clear, which is often referred to as the 'black box' problem. This form of representation stands for both classification and regression, where *y* is a definite class label and a real number, respectively. In **unsupervised learning models**, however, label *y* for output is not required and the models attempt at obtaining patterns directly from the input data *x*. Few examples of biological models employing unsupervised learning are – PCA (principal component analysis), detection of outlier and clustering.

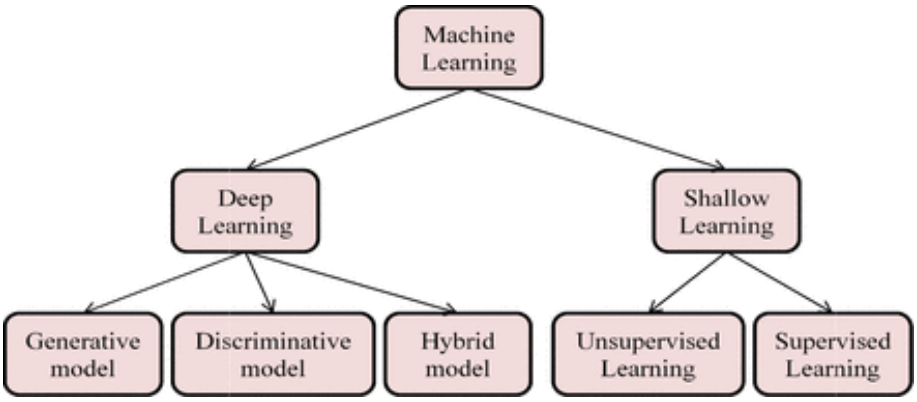


Figure 14: Classification of Machine Learning Techniques. (Image Source: A. Jabeen et al., 2018 [94]).

It proves to be an arduous, and a skillful task to derive input features *x* from raw data. So much so that the process is highly specific to the problem, nevertheless, it is vital for efficient model performance. For data of higher dimensions this is an even bigger problem. Lately, the problem has been tackled by the use of deep artificial neural networks [95] to represent the data, which automate the process. In a deep neural network, data is taken from the input layer and converted into a representation of theoretical features by sequentially and repeatedly accumulating outputs from previous layers in a manner that is information-driven, in the process capturing and formulating complex functions. Deep learning techniques and their variations are widely used today in the domains of natural language processing [96], speech and image recognition [97], and in bioinformatics and computational biology [98], [99].

2.3.2 Artificial Neural Networks

Inspired by biological neural networks in the human brain, an artificial neural network comprises multiple layers of neurons (interconnected mathematical units) [100]. While the width of a neural network signifies the number of neurons in a single layer, the depth indicates the number of hidden layers in the neural network. 'Deep networks' is a term that denotes artificial neural networks, consisting of large number of hidden layers, that can be trained.

The workflow of deep networks is such that first the input data is received in the initial layer, then the data is processed through multiple hidden layers in a non-linear manner and it emerges converted, and finally an output layer computes the data to present (Figure 15 Part A). A neuron in a hidden layer is connected to every single neuron of the previous layer. Every neuron determines a sum of its input data which comes from weights and estimates output *F(x)* by employing a non-linear activation function (Figure 15 Part B). The ReLU (rectified linear unit) sets a limit for negative signals to '0' and it goes through a positive signal; it is most prevalent activation function and is open to learning quickly as opposed to substitute functions, such as tanh unit and sigmoid function [101].

From the samples of inputs and outputs, weights ($w^{(i)}$) are obtained among neurons, as factors, which encapsulate the data as represented by the model. Loss function i.e. *L(w)* measures the extent of fitness of output of the model to original sample label (Figure 15 Part A, down). Training minimizes *L(w)*, however, minimization is difficult due to greater dimensionality and non-convex nature of the loss function, like the crests and troughs in graph (Figure 15 Part C). Decades later it became possible to estimate the loss function gradient through chain rules for derivatives using the 'backward propagation algorithm' [102]. This enabled use of stochastic gradient descent for more effective training of neural networks. Learning involves comparison of true label and predicted label to estimate current model weights loss, then backward propagating that loss through network to estimate loss function gradients and upgradation (Figure 15 Part A). Gradient based descent has been used to optimize loss function *L(w)* in a classical manner. At every stage, the weight vector (red dot), follows down the slope, *dw* signifying the downwards arrow, by learning the rate η (vector length). Various spheres pertaining to the loss function are explored by the decaying of the learning rate with time, such as fine-tuning factors (parameters) with even minute rates of learning in subsequent stages of training of model. Already known and existing

mathematical models could be used for studying deep learning further.

With the advent of artificial neural networks, certain architectures have come to be developed, which serve as alternatives for specific applications. For example, CNN (convoluted neural networks) for image processing, RNN (recurrent neural network) for sequential data [103], or autoencoders [104] or Boltzmann machines [105] for unsupervised learning. Characteristics that enable network architecture are extracted in an information-driven manner by assessing performance of model and that of its validation data set.

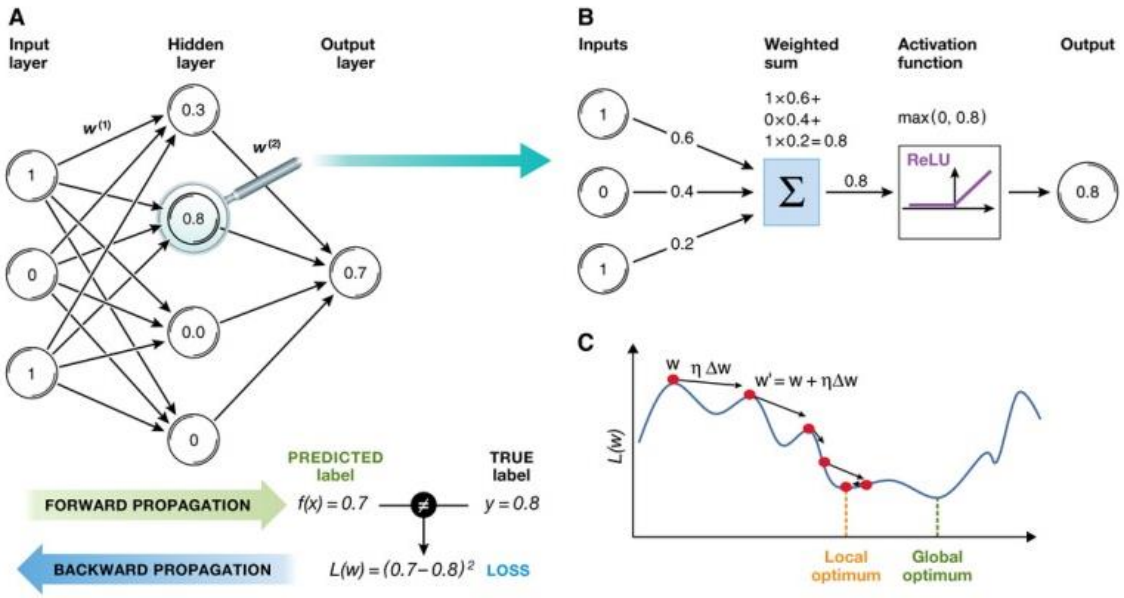


Figure 15: An Artificial Neural Network and its working (Image Source: Angermueller C, et al., 2016 [93])

2.3.3 Convolutional Neural Network

CNNs or Convolutional neural networks were initially greatly used in neuroscience laboratories to study visual cortex of cats. They were invented by Hubel and Wiesel and it was found that felines had a visual cortex characterized by neurons of smaller size that react to smaller subjects in the field of vision, and larger and intricate neurons that react to a bigger subjects [106].

CNNs display data in the form of multidimensional exhibit models, for example, two- dimensional pictures with three channels of different shades [107], [108] or genomic sequences which are unidimensional and contain a single channel per nucleotide [109]. However, the great number of dimensions that the algorithm considers from the input, say for example, a million pixels denoting an image of high resolution: makes it complicated to train a completely linked neural network, since in such a model the number of factors (parameters) would then exceed the quantity of data

used for their fitting. CCNs sidestep this problem by studying the structure of the network and making further assumptions on it, thus constraining the number of factors of import to learn.

A single layer in a CNN consists of a consists of multiple filters, also known as character (feature) maps, or neuron maps; size of a filter equals image dimension from input (Figure 16 Part A). Local connectivity and sharing of parameters – these two incorporations facilitate the reduction in parameters in the model. Each neuron in a feature map is linked only to neighboring patch of neurons of the preceding layer which is known as the 'receptive field'; this is unlike what happens in a completely correlated network. Also, within a feature map all neurons share the same parameters. Therefore, within a feature map all neurons share identical features as those in the preceding layer, albeit at locations which are distinct. For example, in a sequence motif from a sequence in genome, or in an image, different feature maps can detect edges with distinct orientation. By estimating a separate convolution of its receptive field, a particular neuron activity is achieved, which applies an activation function and computes the weighted sum of neurons from input (Figure 16 Part B).

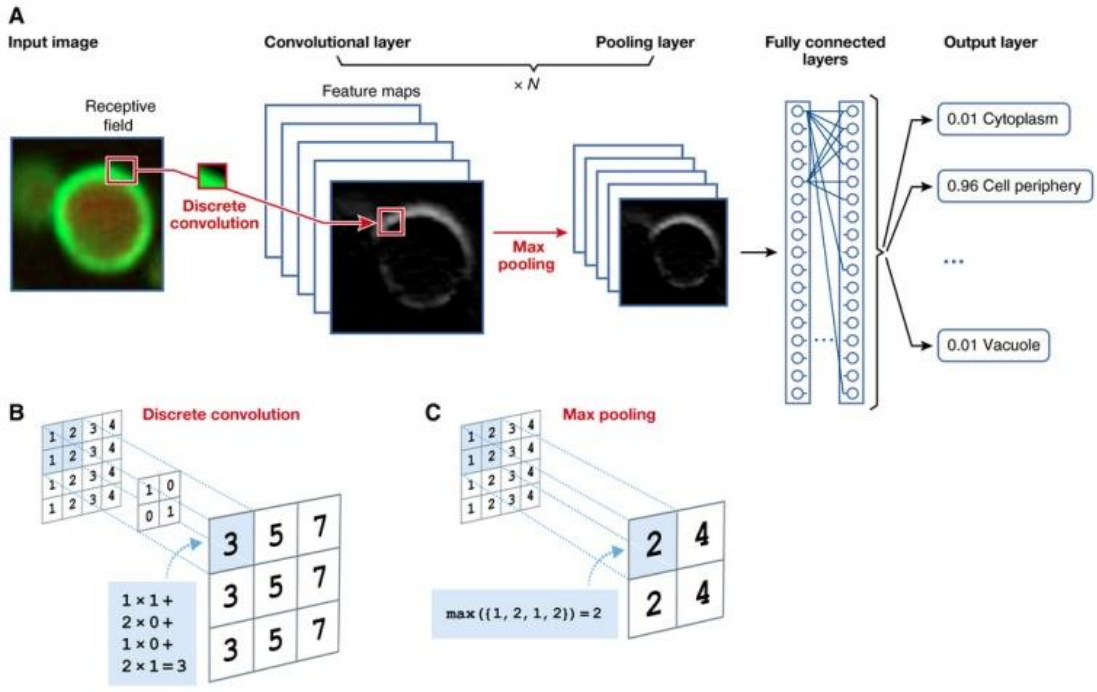


Figure 16: Working of a Convolutional Neural Network (Image Source: Angermueller C, et al., 2016 [93])

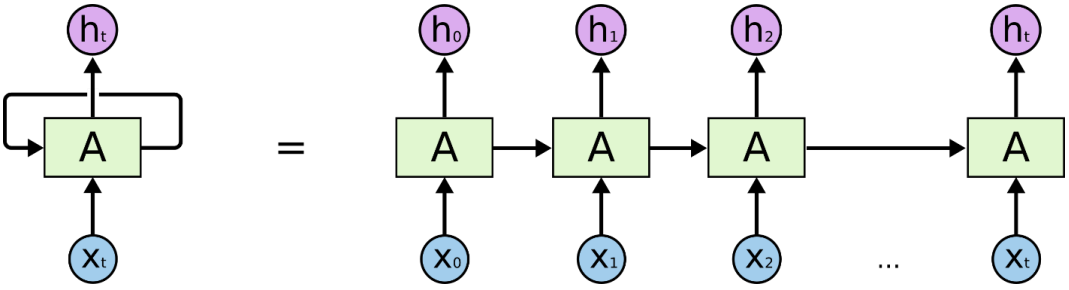
To achieve prediction in the final stage, for example, in identifying objects from an image, the frequency of features and their precise position is insignificant in most applications. The pooling layer makes use of this theory and recapitulates neighboring neurons through computation,

resulting in presentation of feature activities which is smoother (Figure 16 Part C). The image from input is significantly sampled down, when a similar pooling function is applied to small patches of images which are moved by greater than a pixel. This reduces the parameters of the model further.

A classic convolutional neural network therefore comprises numerous convolutional layers and pooling layers; these enable the discovery of increasing number of intangible features at a progressive scale – from small edges to parts of objects, and from them to whole objects. A single layer or multiple linked ones can follow the final pooling layer (Figure 16 Part A). In a model, the super factors i.e. the hyper-parameters, which essentially control the learning process, such as the quantity of feature maps, number of convolutional layers and receptive field size, all depend upon the application and therefore should explicitly be chosen from a dataset for validation.

2.3.4 Recurrent Neural Network

Unlike the human brain, which connect events of previous events while addressing current events, neural networks of the traditional kind (before RNNs) lack this ability. Say, for example a story from a novel is to be classified according to events happening at every page in the novel. Conventional neural networks are unable to identify later events in the pages of novel based upon previous events. Researchers developed Recurrent neural networks to tackle the problem. RNNs contain loops, which allows retention of previous information.



(a) (b)

Figure 17: (a) RNN with loop representation and (b) Unrolled RNN.

Figure 17 (a) represents a part of a neural network (*A*), which takes an input x_t and gives an output of a value h_t. A loop in the diagram enables data transfer from one part of the network to another part. An RNN can be thought of as several copies of one single network, with each copy transferring some information to the next copy.

Figure 17 (b) represents the representation if the loop is unrolled. This structure of RNNs like a chain

indicates that they are closely related to sequences (think, genomic sequences), and lists, and would find major applications by formulating natural architectures in these domains. As a result in the last few decades RNNs have found a range of applications [110], be it in speech and language recognition models, translation, captioning images or computational biology.

The Problem of Long-Term Dependencies

While RNNs can link previous information to current task, the success of such performance is varying depending upon the gap between the relevant information (in the past) and the current task where prediction is needed. It was found that if this gap is extensive, i.e. if the relevant information needed for prediction is in a repeating module of the network that is much in the past then RNN fails. This problem is referred to as the Long-Term dependency problem. In theory, RNNs can handle such problems. However, in practice, they do not seem to be able to learn them. LSTMs and GRUs do not have this problem.

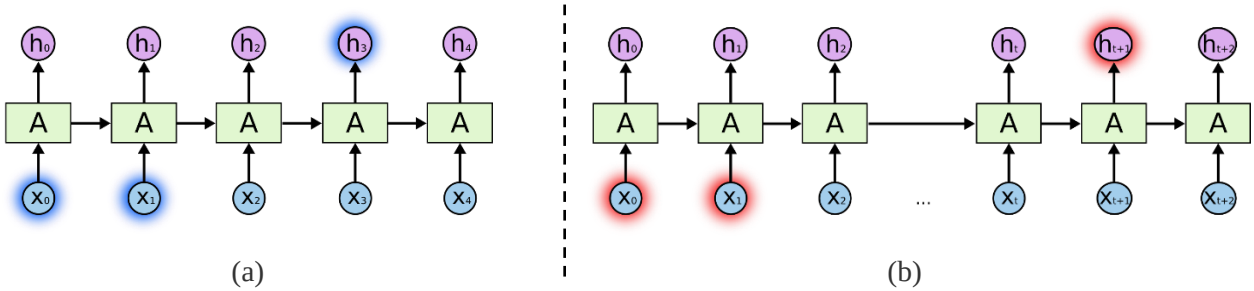


Figure 18: (a) RNN without long-term dependency problem, (b) RNN with long-term dependency problem.

2.3.4.1 Long Short-Term Memory (LSTM) Networks

LSTMs are a type of RNN specifically made for circumventing the long-term dependency problem in traditional RNNs. They were first incorporated by Hochreiter & Schmidhuber in 1997 [111] after which they were popularly used and refined in the years that followed.

LSTMs by default explicitly remember data for long time periods. Like simple RNNs they also have a chain like structure however the repeating module is structurally different – in place of just one layer of neural network (like 'tanh' layer in simple RNN) there are four layers of neural networks interacting in a unique manner.

2.3.4.2 Gated Recurrent Units (GRUs)

GRUs are a variation of LSTMs that were introduced by Cho et al. in 2014 [112]. They are also aimed at ameliorating the problem of long-term dependencies. Gated Recurrent Units do this by

incorporating a forget gate with input gate to form an ‘update gate’, however the number of parameters are fewer than LSTM since an output gate is absent. GRU also merges the hidden and the cell state. It has a simpler architecture than LSTMs and is gaining increasing popularity.

(a)

(b)

(c)

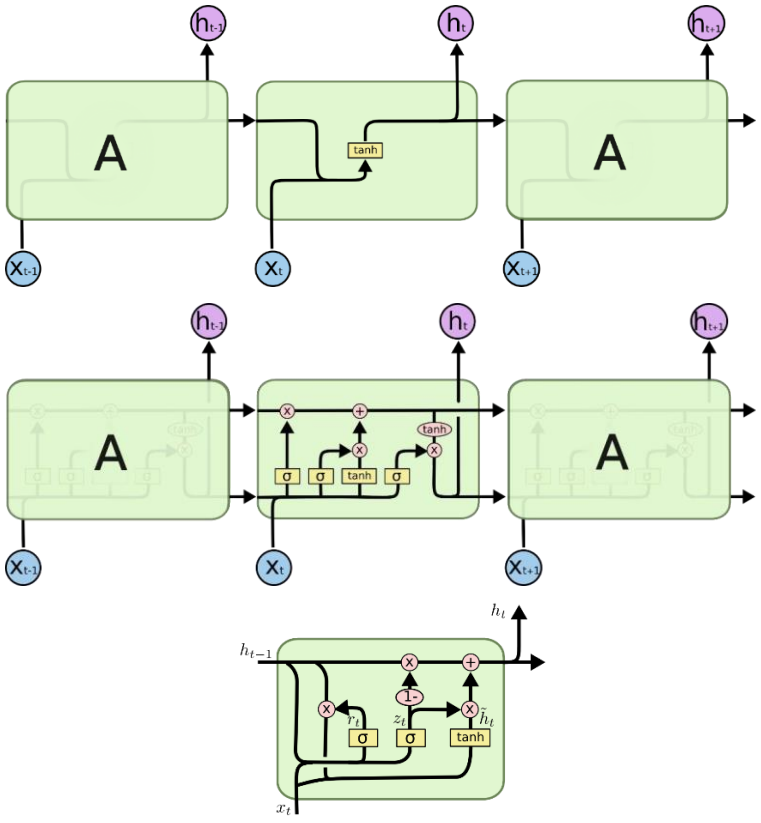


Figure 19: (a) RNN architecture, (b) LSTM architecture, (c) GRU architecture.

CHAPTER 3 METHODOLOGY

3.1 Data Curation

The objective of the study, which was to determine site accessibility (bound or unbound) of nucleotides in RNA by prediction, was achieved through a neural network architecture comprising of embedding, CNN and GRU layers. For building the model known structural data on RNA sequences was required.

Structures of RNA were obtained from two popular RNA secondary structure databases namely RNA STRAND [20] and Comparative RNA Web (CRW) Site [113]. The databases are as described below.

The RNA STRAND database is a comprehensive collection of known RNA secondary structures that have been carefully assembled from trusted databases. It further provides easy online tools that facilitate searching, analyzing, and downloading of data as per selection. RNA STRAND – the "RNA secondary STRucture and statistical ANalysis Database" is publicly available and it consists of total 4666 RNA structures obtained from different databases, as listed in Table 2.

Table 2: Provenience of structures in RNA STRAND

Source database	**Number of RNA structures**
RCSB Protein Data Bank	1059
Gutell Lab CRW Site	1056
tmRNA Database	726
Sprinzl tRNA Database	622
RNase P Database	454
SRP database	383
Rfam database	313
Nucleic Acid Database	53
TOTAL	**4666**

- The list of types of RNA in the database is as follows: Transfer Messenger RNA, 16S Ribosomal RNA, Transfer RNA, Ribonuclease P RNA, Synthetic RNA, Signal Recognition Particle RNA, 23S Ribosomal RNA, 5S Ribosomal RNA, Group I Intron, Hammerhead Ribozyme, Other Ribosomal RNA, Other Ribozyme, Group II Intron, and

Cis-regulatory element. Among these Transfer Messenger RNA, 16S Ribosomal RNA and Transfer RNA are predominant with 726, 723 and 707 entries, respectively.

- In all the 4666 RNA structures different structural motifs were found. They were – pseudoknots, multibranched loops, internal loops, bulge loops, hairpin loops and non-canonical base pairs. Among these the most predominant were hairpin loops which were found in about 4575 RNA structures, with 43442 total occurrences [20].
- The RNA structures are not limited to human beings as source but are attributed to a wide range of organisms for their provenance.
- Link to the database: http://www.rnasoft.ca/strand/

The CRW or the "Comparative RNA Web Site" is a database consisting of wide range of entries of several different kinds of RNA in terms of sequence and structure, obtained through comparative sequence analysis. The database aims to shed light on RNA structural information through analysis of phylogenetic relationships between them.

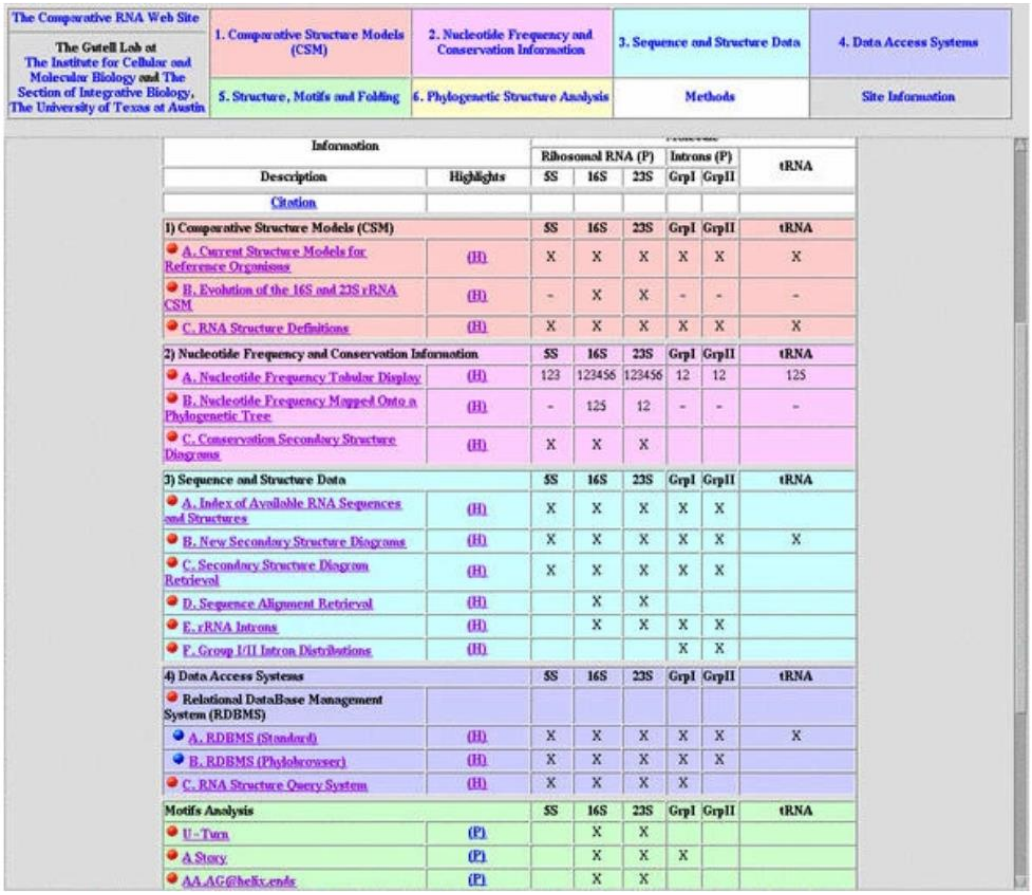


Figure 20: Homepage of CRW site when it was launched in 2002 (Image Source: Cannone, J.J. et al., 2002 [113]).

- The database consists of four key comparative information systems. These are: Current Comparative Structure Models, Sequence and Structure Data, Data Access Systems, and Nucleotide Frequency & Conservation Information.
- The sequence and structural information is from three types of ribosomal RNA, namely, 5S, 16S, and 23S rRNA, two kinds of catalytic intron RNAs (group I, group II) and the adaptor molecule - transfer RNA (tRNA). CRW site predominantly contains 16S rRNA data.
- Link to the website: http://www.rna.icmb.utexas.edu/

For our study, '.ct' files (which secondary structure information for a sequence) were taken from both the databases. All 4666 files were obtained from the RNA STRAND database while 17032 files were obtained from Comparative RNA Web (CRW) Site – all of them containing 16S rRNA structures. Therefore, total 21698 sequences from both databases were collected.

3.2 Data Preparation

3.2.1 Data Cleaning

Of the 21698 sequences taken from both the databases, many consisted of unknown nucleotides. Such sequences were filtered out using Python. Only the sequences with letters 'A', 'U', 'G', and 'C' were retained. This resulted in curated set of 21145 sequences.

3.2.2 Data Clustering

Among the remaining 21145 entries, majority were of 16S rRNA, since about 17032 entries were from the CRW site which were all of the type16S rRNA. Therefore, there was a need to remove sequences which were largely similar and retain only the unique sequences. This was achieved through CD-Hit-EST program.

- CD-HIT is a popular program used for comparing and subsequently clustering nucleotide or protein sequences [114], [115]. The program is extraordinarily fast and can handle exceptionally large databases. It can eliminate bias in datasets and facilitate improved understanding of structure in data, while can greatly reducing manual and computational

efforts in sequence analysis. Popular tools such as Uniprot, SWISS-MODEL, FFAS and CAMERA, all indirectly employ CD-Hit for clustering data.

The CD-Hit-EST clustering performed upon the 21145 RNA entries reduced the number of sequences significantly. CD-Hit clusters sequences according to user define similarity cut-off value. Stringent Parameters used for clustering, including an 85% cut off, are enlisted in Table 3.

Table 3: Parameters for Sequence Clustering.

Parameters	Symbols	Value
Word length	n	12
Global sequence identity	G	0
Length of throw away sequences	L	12
Length difference cut-off in nucleotides	S	0.80
Alignment coverage for the shorter sequence	aS	0.98
Number of threads	T	0

After clustering and reducing sequences with similarity as per Table 3, 4400 sequences were obtained. Among these the minimum sequence length was 13 nucleotides and maximum sequence length was 4379 nucleotides. Median sequence length was 463 and all in all the sequences contained 29.7 million nucleotides. All sequences were padded up to the maximum length, i.e. 4379 with '<pad>' tag before training the model.

3.3 Building the Model

3.3.1 Python for model implementation

Python Keras library (2.4.2) for deep learning, using TensorFlow (1.14) was used in the background for model development and training.

TensorFlow is an open-source programming library widely finding application in dataflow programming across a broad array of possible undertakings. It is a representative math library, and is similarly used for machine learning applications, for example, neural systems. In Google browsing, it finds application in both research and generation.

Keras is a high-level neural networks API, written in Python and capable of running on the top of TensorFlow, CNTK, or Theano. It was developed with the aim of enabling rapid experimentation. Being able to go from an idea to result stage, in minimum time and the least possible delay is crucial for good research. Keras offers simple and rapid prototyping, in terms of - user friendliness, modularity, and extensibility. It supports both CNNs and RNNs, as well as combinations of the two; and runs effortlessly on CPU and GPU.

3.3.2 Steps in building the model

The model was created in such a manner that single nucleotide features, global contributing features, and local window features; can all be extracted.

- Initial step was the introduction of an embedded layer in the model so that it may learn or extract features of single nucleotides.
- As a second step, a one-dimensional CNN layer (Conv1D) was added upon the embedding layer. It was meant to function as a sliding window with user-defined window length to extract locally contributing features.
- On top of the Conv1D layer, a layer of bidirectional Gated Recurrent Units (GRU) was superimposed. GRUs are a type of RNN which contain a separate memory channel to deal with the problem of 'vanishing gradient'; this memory channel contains updated and reset gates for retaining impacts for values far way in the time step. The bidirectional GRU layer would learn and extract global features from both the direction of the sequence.
- The next step was the concatenation of the previous two layers i.e. the integration of the learned features in Cov1D and GRU layers.
- The results from concatenation i.e. all the neuron extract features were fed to a time distributed dense layer of 4379 time-steps and 3 dense layers in each time-step with application of 'SoftMax activation function'.
- Finally, the nucleotide distributed dense layer resulted in an output where the last time-step gives probability of each nucleotide being 'bound', 'free' or a 'pad' (no nucleotide), as the sequences were padded up to the maximum length of 4379. The 'start' and 'pad' tag in the start of a sequence take the maximum length of sequences to 4381 in the network.

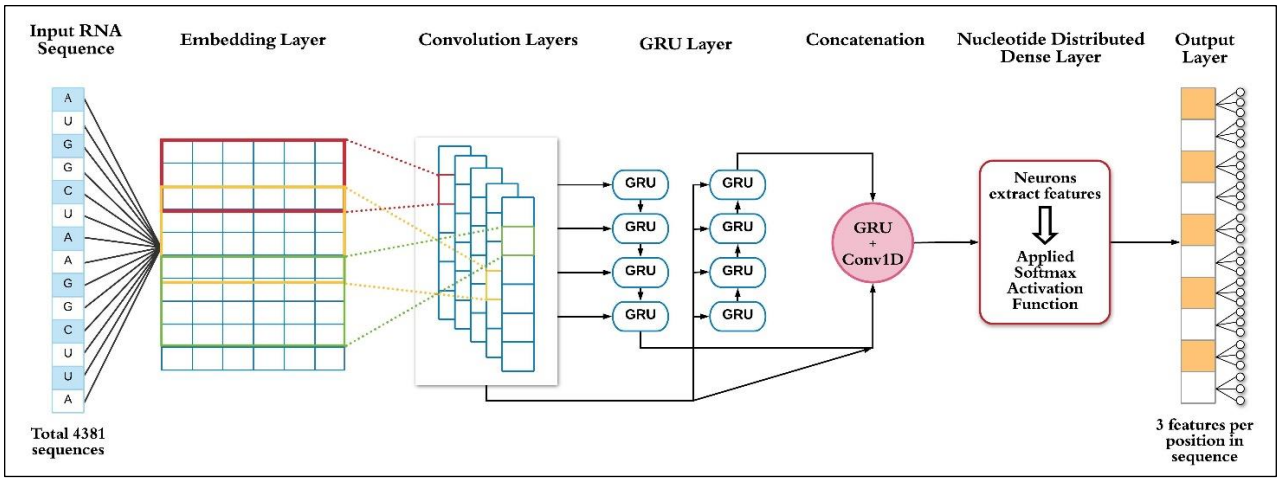


Figure 21: Architecture of the proposed model: an ensemble of Embedding, CNN and GRU layers for extracting features. Extracted features were fed to step distributed dense layer with softmax function for determining the pairing probability of nucleotide.

3.3.3 Neural Network created

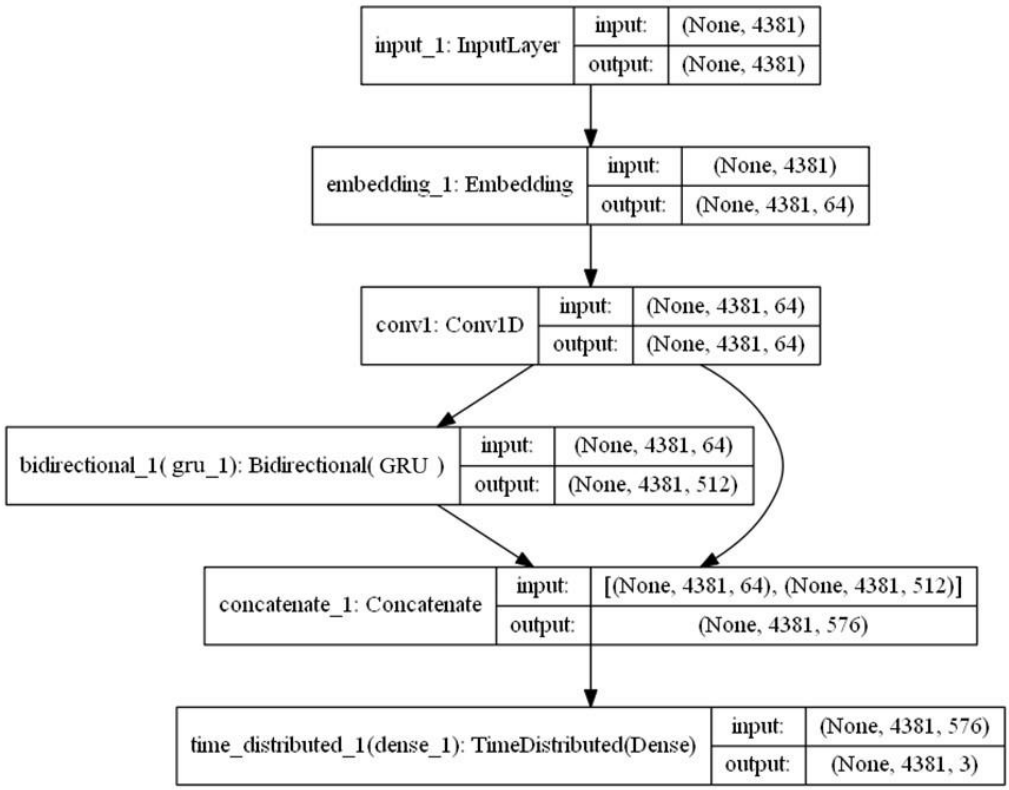


Figure 22: A workflow depicting the neural network model consisting of Embedding, Cov1D and GRU layers.

The python code for the model training and prediction is shown in APPENDIX I and II, respectively.

3.4 Training the Model

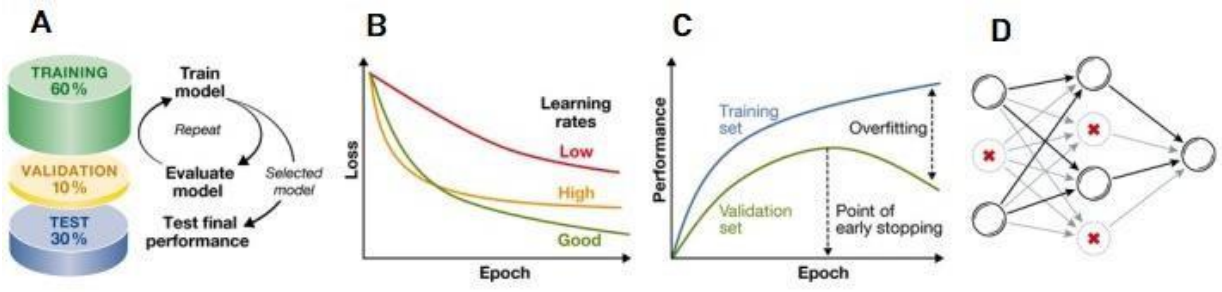


Figure 23: Classic approach for training a neural network model (Image Source: C. Angermueller., 2016 [93])

Training of a machine learning model is done to discover parameters (*w)* which minimize the objective function *L(w),* which measures the fit between model predictions parameterized by *w* and the actual observations. The most common objective functions are mean-squared error for regression and cross-entropy for classification. Minimizing *L(w)* is a challenging task because of its non-convex and high dimensional nature. Refer section 2.3.2 to gain a more elaborate explanation.

3.4.1 Establishing Hyperparameters in the Network

The most challenging part for training a deep learning model is to decide the hyperparameters such as the number of neurons in layers, features to extract, window length and filters of CNN etc. A lower number of neurons will not be able to learn all the features and the model will not perform well, on the other hand, a larger number of neurons tend to memorize the training dataset leading to overfitting. For deciding the hyperparameters, we used a grid search method on various combinations of the above stated hyperparameters and selected the combination which showed the best performance on the validation dataset. It was found that 64 features per nucleotide in the embedding layer, a window size of 100 and 64 filters in 1D CNN, and bidirectional GRU with 256 recurrent units, all perform best on the validation set.

3.4.2 Apportioning Data into Training, Validation and Testing

Machine learning models need to be not just trained but also must be validated and tested on independent sets of data to avoid overfitting and assure that the model will perform well on

unknown data. This is the standard practice for deep neural networks. The training set is used by the model to learn with different hyper-parameters imposed, which the validation set evaluates later. Model giving the greatest execution , such as 'prediction accuracy' or 'mean-squared error', is chose; it is additionally assessed on the test dataset to calculate its execution on unknown data and in order to compare to other methods. Out of 4379 data files we used 70% for training, 20% for validation and 10% for testing, i.e. 3080 sequences for training, 880 for validation and rest 440 for testing.

3.4.3 Learning Rate and Batch size

The clump size and learning rate of a model should be picked with deliberation since their values can largely affect its training rate and performance. Different rates of learning are employed in a typical model, such as, 0.1, 0.01 or 0.001, with 0.01 considered as default. Most common clump size for applications is of 128 samples for training. Increasing batch size accelerates training while a reduced batch size would decrease utilization of memory. Classical models work well with the relation of smaller batch size with larger rate of learning and vice-versa. In our work we used a default learning rate of 0.01 with a minimum learning rate of 0.00001 and a batch size of 64.

3.4.4 Avoiding Overfitting

Overfitting is quite a common problem that occurs while attempting to create deep neural networks. A major contributing factor to this is the non-linear nature of the network with multiple parameters. As a result, an overly complex model that is corresponding to the size of the training data can lead to overfitting. The problem can be tackled by either increasing the training data size, such as, through augmentation, or by reducing the model complexity, for example by decreasing the number of units and hidden layers. In our model we had a 50% dropout rate between layers to prevent overfitting. 'Early stopping' function has also been applied which stops training when validation accuracy is consistent, in maximum 3 epochs, with an increasing training accuracy. Furthermore, learning rate reduces ('reduced lr') when there is no increase in validation accuracy in subsequent epochs.

3.4.5 Technical Specifications

The model was trained on a server with configuration 32GB RAM, 11GB NVIDIA Tesla K80

GPU and Intel Xeon 8 Cores CPU with Adam optimizer, Categorical-crossentropy loss function, learning rate decomposition and early stopping method. Learning rate decomposition decreases the learning rate of the optimizer whenever the model halts improving on the training set. Early stopping technique stops the training of the model when the model tends to overfit by comparing the training and validation accuracies.

3.5 Model Evaluation

Model was evaluated on the test set in terms of Area under ROC curve, sensitivity, specificity, and Matthews correlation coefficient which are the well-accredited measurements for evaluation of any classification-based model. The proposed model was compared with other states of art models by evaluating their sensitivity and precision on test data.

3.6 Predicting Site Accessibility of RNA Secondary Structures

After training for 50 epochs overfitting was successfully curtailed and model showed sufficient training and validating to be tested on test data. The model was optimized to predict the probability of a nucleotide being ‘bound’ or ‘free’. Results in this regard gave a high Q2 accuracy and amicable results as shown in the next section.

CHAPTER 4 RESULTS

- After training and validation, the model was tested on test data and its Q2 accuracy was calculated. Q2 accuracy is a two state per residue accuracy which measures the percentage of precisely anticipated nucleotides in all the three classes. It is given by the sum C_i/N which is the ratio of number of precisely predicted nucleotides in a class (C_i) to the total number of nucleotides (N). Q2 accuracy is the optimized test accuracy that is correctly representative of probability of site accessibility of a nucleotide, from the given sequence, after removing 'pad' regions. The model acquired a Q2 accuracy of 0.89 on the test data.

- The Receiver Operating Characteristic curve is a classifier estimator that is produced by plotting the true positive rate (TPR) against the false positive rate (FPR). The TPR is a representation of the 'probability of detection'. It can also be referred as sensitivity or recall. FPR on the other hand is the 'probability of false alarm' in the model and is also known as 1-specificity. The model achieved a ROC of 0.90 (Figure 23) for both class 0 and class 1. Here class 0 denotes residues which are bound (B), while class 1 denotes residues which are free (F).

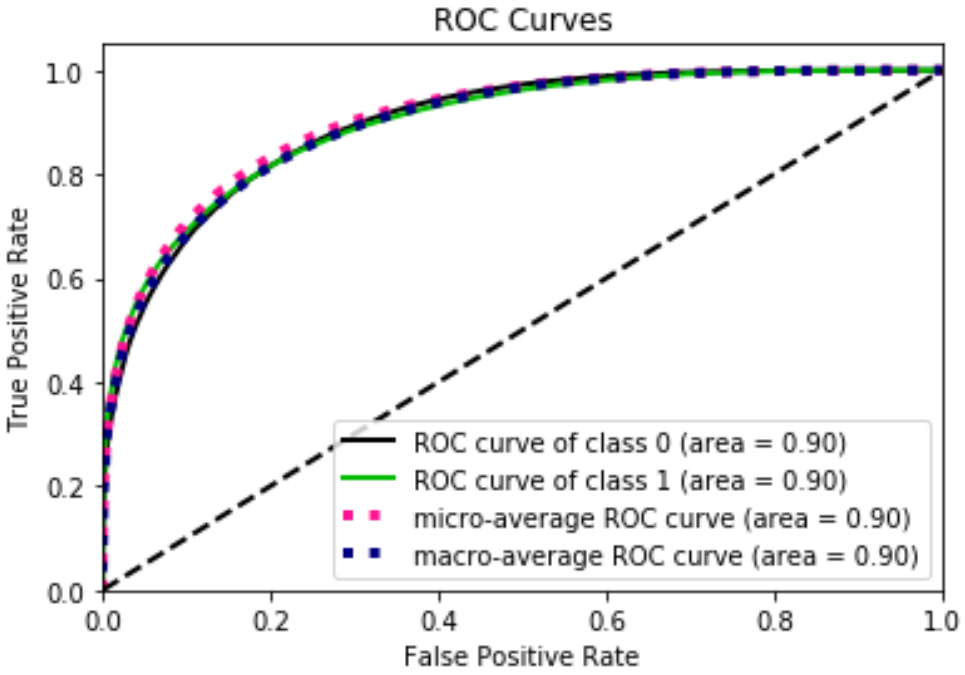


Figure 24: ROC curve of the proposed model on test set; class 0 is 'B' and class 1 is 'F'.

- A Precision-Recall Curve represents the tradeoff between precision and recall. The higher the area under the curve, the higher is the recall and precision. High precision score translates to a low false positive rate, while a high recall score relates to a low false negative rate. Higher scores for both assure that the classifier is returning accurate results (indicated by high precision), as well as returning a majority of all positive results (indicated by high recall). On calculating Precision-Recall Curve for our model it was seen that area under the curve for predicting a residue that is free (class 1) was 0.87, whereas for residue that is bound (class 0), area under the curve was 0.92 (Figure 25).

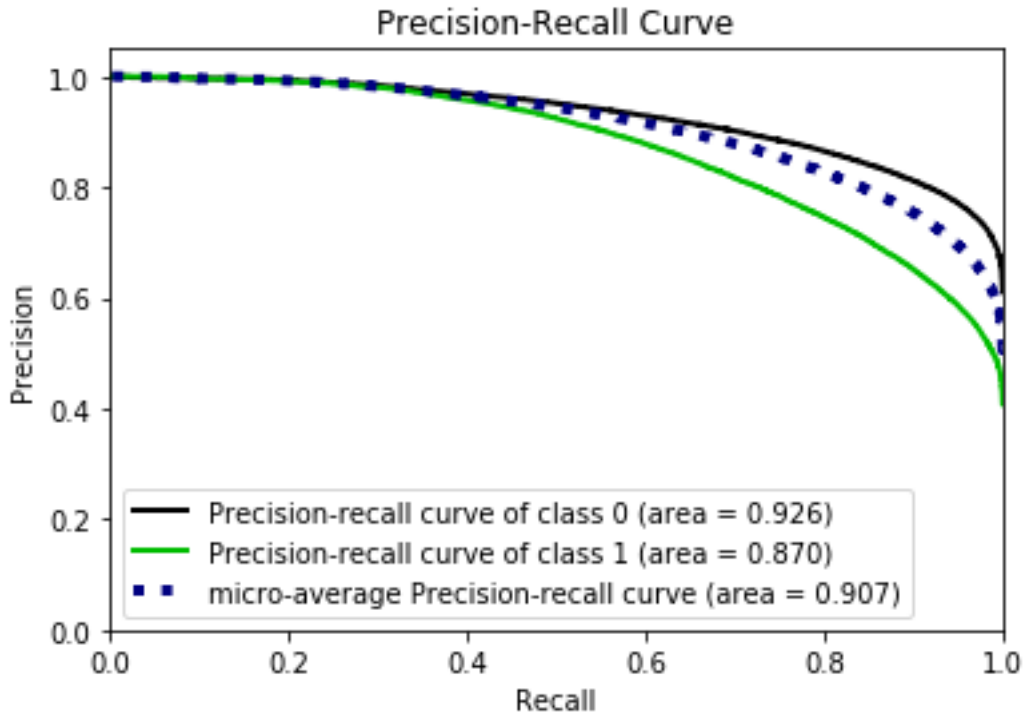


Figure 25: Precision-Recall curve of the proposed model on test set; class 0 is 'B' and class 1 is 'F'.

- Table 4 represents the model's diagnostic capability upon testing on test set in terms of sensitivity, specificity, precision, and Matthews correlation coefficient (MCC). The values for these are 0.75, 0.79, 0.78, and 0.55, respectively.

Table 4: Evaluation of the model on test data.

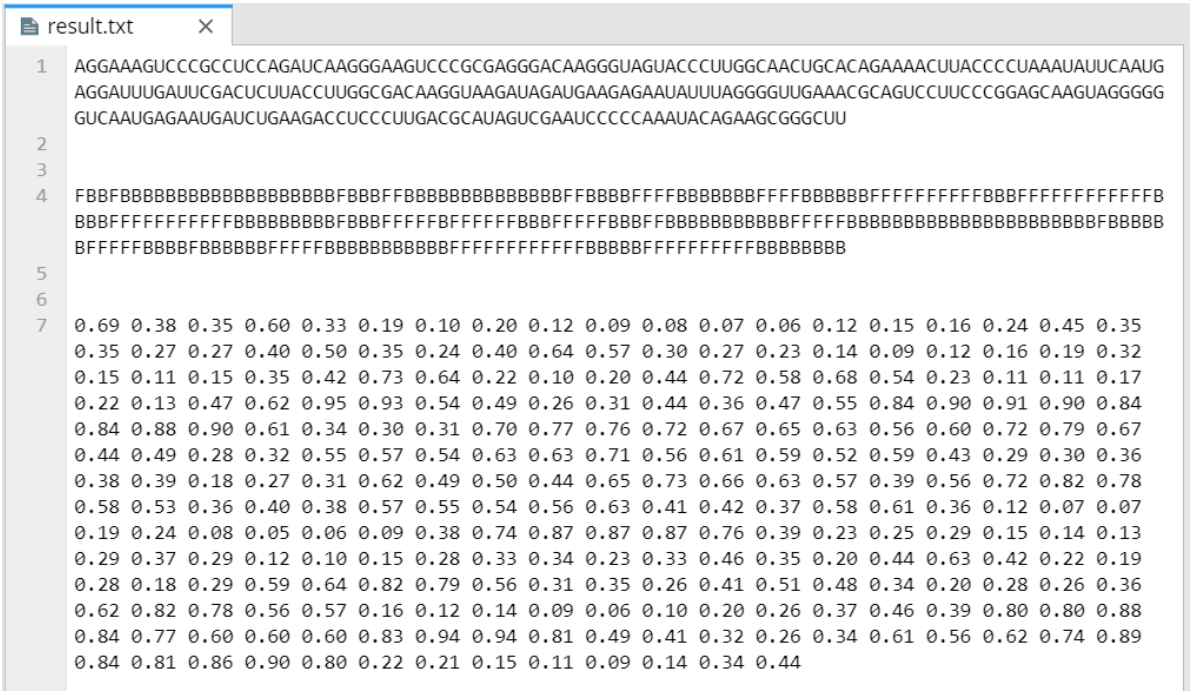
ROC	Sensitivity	Specificity	Precision	MCC
0.90	0.75	0.79	0.78	0.55

- The proposed model was compared with other state-of-the-art prediction tools and servers in terms of sensitivity and precision, shown in table 5. The proposed model outperforms other models with a significant margin.

Table 5: Comparison of the proposed model with various state of the art models on test dataset.

Method	Sensitivity	Precision	Reference
RNAshapes	0.635	0.57	[116]
Mfold	0.64	0.57	[32]
RNAfold	0.625	0.545	[36]
RNAstructure	0.675	0.595	[38]
Proposed Model	**0.75**	**0.78**	

- Upon running the model for a sequence an output file is generated named "results.txt" containing the input RNA Sequence, predicted free residues ('F') and bound residues ('B') and the probability of each position of nucleotide being free. Figure 26 depicts an example output file.


result.txt

```
AGGAAAGUCCCGCCUCCAGAUCAAGGGAAGUCCCGCGAGGGACAAGGGUAGUACCCUUGGCAACUGCACAGAAAACUUACCCCUAAAUAUUCAAUG
AGGAUUUGAUUCGACUCUUACCUUGGCGACAAGGUAAGAUAGAUGAAGAGAAUAUUUAGGGGUUGAAACGCAGUCCUUCCCGGAGCAAGUAGGGGG
GUCAAUGAGAAUGAUCUGAAGACCUCCCUUGACGCAUAGUCGAAUCCCCCAAAUACAGAAGCGGGCUU


FBBFBBBBBBBBBBBBBBBBBBFBBBFFBBBBBBBBBBBBBBFFBBBBFFFFBBBBBBBFFFFBBBBBBFFFFFFFFFFBBBFFFFFFFFFFFFB
BBBFFFFFFFFFFFBBBBBBBBBFBBBFFFFFBFFFFFFBBBFFFFFBBBFFBBBBBBBBBBBFFFFBBBBBBBBBBBBBBBBBBBBBFBBBBB
BFFFFFBBBBFBBBBBBFFFFFBBBBBBBBBBBFFFFFFFFFFFFBBBBBFFFFFFFFFFBBBBBBBB


0.69 0.38 0.35 0.60 0.33 0.19 0.10 0.20 0.12 0.09 0.08 0.07 0.06 0.12 0.15 0.16 0.24 0.45 0.35
0.35 0.27 0.27 0.40 0.50 0.35 0.24 0.40 0.64 0.57 0.30 0.27 0.23 0.14 0.09 0.12 0.16 0.19 0.32
0.15 0.11 0.15 0.35 0.42 0.73 0.64 0.22 0.10 0.20 0.44 0.72 0.58 0.68 0.54 0.23 0.11 0.11 0.17
0.22 0.13 0.47 0.62 0.95 0.93 0.54 0.49 0.26 0.31 0.44 0.36 0.47 0.55 0.84 0.90 0.91 0.90 0.84
0.84 0.88 0.90 0.61 0.34 0.30 0.31 0.70 0.77 0.76 0.72 0.67 0.65 0.63 0.56 0.60 0.72 0.79 0.67
0.44 0.49 0.28 0.32 0.55 0.57 0.54 0.63 0.63 0.71 0.56 0.61 0.59 0.52 0.59 0.43 0.29 0.30 0.36
0.38 0.39 0.18 0.27 0.31 0.62 0.49 0.50 0.44 0.65 0.73 0.66 0.63 0.57 0.39 0.56 0.72 0.82 0.78
0.58 0.53 0.36 0.40 0.38 0.57 0.55 0.54 0.56 0.63 0.41 0.42 0.37 0.58 0.61 0.36 0.12 0.07 0.07
0.19 0.24 0.08 0.05 0.06 0.09 0.38 0.74 0.87 0.87 0.87 0.76 0.39 0.23 0.25 0.29 0.15 0.14 0.13
0.29 0.37 0.29 0.12 0.10 0.15 0.28 0.33 0.34 0.23 0.33 0.46 0.35 0.20 0.44 0.63 0.42 0.22 0.19
0.28 0.18 0.29 0.59 0.64 0.82 0.79 0.56 0.31 0.35 0.26 0.41 0.51 0.48 0.34 0.20 0.28 0.26 0.36
0.62 0.82 0.78 0.56 0.57 0.16 0.12 0.14 0.09 0.06 0.10 0.20 0.26 0.37 0.46 0.39 0.80 0.80 0.88
0.84 0.77 0.60 0.60 0.60 0.83 0.94 0.94 0.81 0.49 0.41 0.32 0.26 0.34 0.61 0.56 0.62 0.74 0.89
0.84 0.81 0.86 0.90 0.80 0.22 0.21 0.15 0.11 0.09 0.14 0.34 0.44
```

Figure 26: Output file of model with probability of being free and annotation 'B' and 'F' per nucleotide position.

CHAPTER 5 DISCUSSION AND CONCLUSION

Previous works on RNA site accessibility prediction were largely based on comparative sequence analysis, experimentally measured thermodynamic parameters or those based on stochastic context free grammars (SCFE) and more lately based on either of local or global folding machine learning-based algorithms. The current method providing probability of site accessibility with higher accuracy is an improvement over these methods. It utilizes both local features (CNN Layer) and global features (GRU layer) while optimizing on their individual drawbacks and at the same time it also considers all types of RNA folding, including pseudoknots and non-canonical base pairing. Use of RNN layers in probabilistic models, like GRUs, may offer greater accuracy since they tackle the issue of long-term dependency, which the RNA secondary prediction methods can greatly benefit from. Moreover, unlike one-hot encoding-based machine learning algorithms for inputs, the Embedded-CNN-GRU based algorithm in the current model positions nearest neighbours in the embedding space. This helps cluster them into categories and to visualize relations between those categories.

However, the proposed model has specific and generic limitations that must be addressed. While the model quite precisely ascertains the bound, unbound nature of a nucleotide site, it does not predict with which nucleotides the bound site interacts and in what manner. A model capable of predicting this would shed light on the RNA secondary structure and its conformations directly predicted from the sequence; this is the scope of current model. In machine learning models, biasness to training data leads to overfitting – which in turn decreases the veracity of the predictions. Overfitting has been known to exist in many models, including ContextFold [48]. A reason for this is also because of the dominance of RNA databases with a single RNA type. While a 50% dropout rate has been incorporated in the proposed model to deal with overfitting, it's also important to understand that the limited diversity of well-curated single-sequence RNA secondary structures contributes to the problem and prevents improvement in statistical models for RNA secondary structure prediction. Furthermore, there is a longstanding need for a greater number of experimentally identified RNA structures with high-resolution – a number which would be optimum to train probabilistic models and further increase their efficacy. In retrospect, a futuristic model with sufficient experimentally determined RNA secondary structures and a robust probabilistic model for predicting unknown structures could be ideal. Finally, the universal

problem of 'black box' in machine learning models prevents an understanding of how features are extracted till the output layer, which robs one of the potentials of improving the models from a more biological perspective.

Predicting RNA site accessibility would find several applications with the academia working on RNA. It would help predict sites for potentially binding of ribosome, on the mRNA which would open avenues for disease regulation. It would also help determine protein binding sites on the mRNA which play a role in post-transcriptional gene regulation, translation, and splicing. Furthermore, the study would enable prediction of RNA sites playing a role in RNA interference, namely miRNA binding sites in Eukaryotes and siRNA binding sites in Prokaryotes. The knowledge of the pairing probability of RNA nucleotides would further add to the bigger goal of RNA secondary structure prediction with higher accuracy [42]. RNA therapy, involving drugs designed to act on specific RNAs, is another area where RNA site accessibility has much to contribute to.

The scope of the present study encompasses prediction of RNA site pair probability by employing an ensemble of embedding, 1D CNN and GRU layers. This would help in establishing the positions at which RNA nucleotides are participating in self-folding, forming loops, stems, and pseudoknots. However, subsequent work on predicting which residues among A, U, G and C are binding at these positions would greatly help the research community in the field. This vision from the current study can be carried forward.

Additionally, exploring hybrid methods which make use of computational methods (probabilistic algorithms) and other techniques such as RNA chemical probing, dynamic programming, or comparative methods, could offer novel solutions and increment in prediction accuracy. This could help create reactivity profiles for RNA structures which combined with computational algorithms may help in predicting RNA structure more accurately. In the long run RNA secondary structure prediction has much potential in contributing to gaining insights into RNA-disease relationships, thrusting RNA biology towards improving human healthcare.

CHAPTER 6 REFERENCES

[1] H. Zhang et al., "A new method of RNA secondary structure prediction based on convolutional neural network and dynamic programming," Front. Genet., 2019.

[2] R. R. Breaker and G. F. Joyce, "The expanding view of RNA and DNA function," Chemistry and Biology. 2014.

[3] J. Brosius and C. A. Raabe, "What is an RNA? A top layer for RNA classification," RNA Biol., 2016.

[4] F. Crick, "Central dogma of molecular biology," Nature, 1970.

[5] E. V. Koonin, "Does the central dogma still stand?," Biol. Direct, 2012.

[6] R. R. Breaker, "Riboswitches and the RNA world," Cold Spring Harb. Perspect. Biol., 2012.

[7] S. H. Bernhart, "RNA structure prediction," Methods Mol. Biol., 2011.

[8] K. Inamura, "Major Tumor Suppressor and Oncogenic Non-Coding RNAs: Clinical Relevance in Lung Cancer," Cells, 2017.

[9] K. N. Holohan, D. K. Lahiri, B. P. Schneider, T. Foroud, and A. J. Saykin, "Functional microRNAs in Alzheimer's disease and cancer: Differential regulation of common mechanisms and pathways," Frontiers in Genetics. 2013.

[10] C. Battaglia et al., "Candidate genes and MiRNAs linked to the inverse relationship between cancer and alzheimer's disease: Insights from data mining and enrichment analysis," Front. Genet., 2019.

[11] S. R. Eddy, "Non-coding RNA genes and the modern RNA world," Nature Reviews Genetics. 2001.

[12] Z. F. Burton, "The RNA World," in Evolution Since Coding, 2018.

[13] J. Nowakowski and I. Tinoco, "RNA structure and stability," Semin. Virol., 1997.

[14] L. Chen and G. Wong, "Transcriptome Informatics," in Encyclopedia of Bioinformatics and Computational Biology, Elsevier, 2019, pp. 324–340.

[15] N. Amodio et al., "MALAT1: A druggable long non-coding RNA for targeted anti-cancer approaches," Journal of Hematology and Oncology. 2018.

[16] B. Swinnen, W. Robberecht, and L. Van Den Bosch, "RNA toxicity in non-coding repeat expansion disorders," EMBO J., 2020.

[17] K. Darty, A. Denise, and Y. Ponty, "VARNA: Interactive drawing and editing of the RNA secondary structure," Bioinformatics, 2009.

[18] S. Zakov, Y. Goldberg, M. Elhadad, and M. Ziv-Ukelson, “Rich parameterization improves RNA structure prediction,” J. Comput. Biol., 2011.

[19] N. Kim, K. N. Fuhr, and T. Schlick, “Graph applications to RNA structure and function,” in Biophysics of RNA Folding, 2013.

[20] M. Andronescu, V. Bereg, H. H. Hoos, and A. Condon, “RNA STRAND: The RNA secondary structure and statistical analysis database,” BMC Bioinformatics, 2008.

[21] E. Westhof and V. Fritsch, “RNA folding: Beyond Watson-Crick pairs,” Structure. 2000.

[22] E. Rivas, “The four ingredients of single-sequence RNA secondary structure prediction. A unifying perspective,” RNA Biol., 2013.

[23] J. Singh, J. Hanson, K. Paliwal, and Y. Zhou, “RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning,” Nat. Commun., 2019.

[24] M. Kertesz et al., “Genome-wide measurement of RNA secondary structure in yeast,” Nature, 2010.

[25] Y. Ding, Y. Tang, C. K. Kwok, Y. Zhang, P. C. Bevilacqua, and S. M. Assmann, “In vivo genome-wide profiling of RNA secondary structure reveals novel regulatory features,” Nature, 2014.

[26] K. A. Wilkinson et al., “High-throughput SHAPE analysis reveals structures in HIV-1 genomic RNA strongly conserved across distinct biological states,” PLoS Biol., 2008.

[27] I. V. Novikova, A. Dharap, S. P. Hennelly, and K. Y. Sanbonmatsu, “3S: Shotgun secondary structure determination of long non-coding RNAs,” Methods, 2013.

[28] R. Nussinov, G. Pieczenik, J. R. Griggs, and D. J. Kleitman, “Algorithms for Loop Matchings,” SIAM J. Appl. Math., 1978.

[29] M. Zuker and P. Stiegler, “Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information,” Nucleic Acids Res., 1981.

[30] W. Salser, “Globin mRNA sequences: analysis of base pairing and evolutionary implications,” Cold Spring Harb. Symp. Quant. Biol., 1977.

[31] J. Rogers, P. Clarke, and W. Salser, “Sequence analysis of cloned cDNA encoding part of an immunoglobulin heavy chain,” Nucleic Acids Res., vol. 6, no. 10, pp. 3305–3322, 1979.

[32] M. Zuker, “Mfold web server for nucleic acid folding and hybridization prediction,” Nucleic Acids Res., 2003.

[33] N. R. Markham and M. Zuker, “DINAMelt web server for nucleic acid melting prediction,” Nucleic Acids Res., 2005.

[34] N. R. Markham and M. Zuker, “UNAFold: Software for nucleic acid folding and hybridization,” Methods Mol. Biol., 2008.

[35] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster, “Fast folding and comparison of RNA secondary structures,” Monatshefte für Chemie Chem. Mon., 1994.

[36] R. Lorenz et al., “ViennaRNA Package 2.0,” Algorithms Mol. Biol., 2011.

[37] D. H. Mathews, M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker, and D. H. Turner, “Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure,” Proc. Natl. Acad. Sci. U. S. A., 2004.

[38] J. S. Reuter and D. H. Mathews, “RNAstructure: Software for RNA secondary structure prediction and analysis,” BMC Bioinformatics, 2010.

[39] S. Janssen and R. Giegerich, “The RNA shapes studio,” Bioinformatics, 2015.

[40] Z. J. Lu, J. W. Gloor, and D. H. Mathews, “Improved RNA secondary structure prediction by maximizing expected pair accuracy,” RNA, 2009.

[41] B. Knudsen and J. Hein, “RNA secondary structure prediction using stochastic context-free grammars and evolutionary history,” Bioinformatics, 1999.

[42] B. Knudsen and J. Hein, “Pfold: RNA secondary structure prediction using stochastic context-free grammars,” Nucleic Acids Res., 2003.

[43] D. Sankoff, “SIMULTANEOUS SOLUTION OF THE RNA FOLDING ALIGNMENT AND PROTOSEQUENCE PROBLEMS.,” SIAM J. Appl. Math., 1985.

[44] J. Allali and M. F. Sagot, “A new distance for high level RNA secondary structure comparison,” in IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2005.

[45] I. Kalvari et al., “Non-Coding RNA Analysis Using the Rfam Database,” Curr. Protoc. Bioinforma., 2018.

[46] Y. J. Hu, “GPRM: A genetic programming approach to finding common RNA secondary structure elements,” Nucleic Acids Res., 2003.

[47] D. Zhang, X., Deng, Z., and Song, “Neural network approach to predict RNA secondary structures,” J. Tsinghua Univ., vol. 10:038., 2206.

[48] S. Wang, J. Peng, J. Ma, and J. Xu, “Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields,” Sci. Rep., 2016.

[49] C. B. Do, D. A. Woods, and S. Batzoglou, “CONTRAfold: RNA secondary structure prediction without physics-based models,” in Bioinformatics, 2006.

[50] B. Fürtig, C. Richter, J. Wöhnert, and H. Schwalbe, “NMR spectroscopy of RNA,” ChemBioChem. 2003.

[51] K. Sato, M. Hamada, K. Asai, and T. Mituyama, “CentroidFold: A web server for RNA secondary structure prediction,” Nucleic Acids Res., 2009.

[52] I. L. Hofacker, “Vienna RNA secondary structure server,” Nucleic Acids Res., 2003.

[53] Y. Ding and C. E. Lawrence, “A statistical sampling algorithm for RNA secondary structure prediction,” Nucleic Acids Res., 2003.

[54] Y. Ding, C. Y. Chan, and C. E. Lawrence, “Sfold web server for statistical folding and rational design of nucleic acids,” Nucleic Acids Res., 2004.

[55] Y. Ding, “Statistical and Bayesian approaches to RNA secondary structure prediction,” RNA. 2006.

[56] M. Andronescu, A. Condon, H. H. Hoos, D. H. Mathews, and K. P. Murphy, “Computational approaches for RNA energy parameter estimation,” RNA, 2010.

[57] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner, “Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure,” J. Mol. Biol., 1999.

[58] I. Tinoco, O. C. Uhlenbeck, and M. D. Levine, “Estimation of secondary structure in ribonucleic acids,” Nature, 1971.

[59] I. Tinoco et al., “Improved estimation of secondary structure in ribonucleic acids,” Nature New Biology. 1973.

[60] S. Griffiths-Jones, S. Moxon, M. Marshall, A. Khanna, S. R. Eddy, and A. Bateman, “Rfam: Annotating non-coding RNAs in complete genomes,” Nucleic Acids Res., 2005.

[61] Y. Wexler, C. Zilberstein, and M. Ziv-Ukelson, “A study of accessible motifs and RNA folding complexity,” in Journal of Computational Biology, 2007.

[62] R. Backofen, D. Tsur, S. Zakov, and M. Ziv-Ukelson, “Sparse RNA folding: Time and space efficient algorithms,” in Journal of Discrete Algorithms, 2011.

[63] M. Collins, “Discriminative training methods for hidden Markov models,” 2002.

[64] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, “Online passive-aggressive algorithms,” J. Mach. Learn. Res., 2006.

[65] R. Kierzek, M. E. Burkard, and D. H. Turner, “Thermodynamics of single mismatches in RNA duplexes,” Biochemistry, 1999.

[66] M. G. Seetin and D. H. Mathews, “RNA structure prediction: An overview of methods,” Methods in Molecular Biology. 2012.

[67] X. Xu and S.-J. Chen, “Physics-based RNA structure prediction,” Biophys. Reports, 2015.

[68] J. Reeder and R. Giegerich, “Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics,” BMC Bioinformatics, 2004.

[69] S. Bellaousov and D. H. Mathews, “ProbKnot: Fast prediction of RNA secondary structure including pseudoknots,” RNA, 2010.

[70] K. Sato, Y. Kato, M. Hamada, T. Akutsu, and K. Asai, “IPknot: Fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming,” Bioinformatics, 2011.

[71] H. L. Chen, A. Condon, and H. Jabbari, “An O(n5) algorithm for MFE prediction of kissing hairpins and 4-chains in nucleic acids,” J. Comput. Biol., 2009.

[72] E. Rivas and S. R. Eddy, “A dynamic programming algorithm for RNA structure prediction including pseudoknots,” J. Mol. Biol., 1999.

[73] M. S. Andronescu, C. Pop, and A. E. Condon, “Improved free energy parameters for RNA pseudoknotted secondary structure prediction,” RNA, 2010.

[74] H. Jabbari, I. Wark, C. Montemagno, and S. Will, “Knotty: Efficient and accurate prediction of complex RNA pseudoknot structures,” Bioinformatics, 2018.

[75] M. Parisien and F. Major, “The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data,” Nature, 2008.

[76] C. Höner zu Siederdissen, S. H. Bernhart, P. F. Stadler, and I. L. Hofacker, “A folding algorithm for extended RNA secondary structures,” Bioinformatics, 2011.

[77] A. O. Harmanci, G. Sharma, and D. H. Mathews, “TurboFold: Iterative probabilistic estimation of secondary structures for multiple RNA sequences,” BMC Bioinformatics, 2011.

[78] M. F. Sloma and D. H. Mathews, “Base pair probability estimates improve the prediction accuracy of RNA non-canonical base pairs,” PLoS Comput. Biol., 2017.

[79] J. Lee et al., “RNA design rules from a massive open laboratory,” Proc. Natl. Acad. Sci. U. S. A., 2014.

[80] R. B. Lyngsø, J. W. J. Anderson, E. Sizikova, A. Badugu, T. Hyland, and J. Hein, “Frnakenstein: Multiple target inverse RNA folding,” BMC Bioinformatics, 2012.

[81] J. A. Garcia-Martin, P. Clote, and I. Dotu, “RNAiFOLD: A constraint programming algorithm for rna inverse folding and molecular design,” J. Bioinform. Comput. Biol., 2013.

[82] V. Reinharz, Y. Ponty, and J. Waldispühl, “A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution,” in Bioinformatics, 2013.

[83] A. Taneda, “Multi-objective optimization for RNA design with multiple target secondary structures,” BMC Bioinformatics, 2015.

[84] A. Churkin, M. D. Retwitzer, V. Reinharz, Y. Ponty, J. Waldispühl, and D. Barash, “Design of RNAs: Comparing programs for inverse RNA folding,” Brief. Bioinform., 2018.

[85] J. Franklin, “The elements of statistical learning: data mining, inference and prediction,” Math. Intell., 2005.

[86] K. P. Murphy, Machine learning: a probabilistic perspective (adaptive computation and machine learning series). 2012.

[87] J. Li, T. Ching, S. Huang, and L. X. Garmire, “Using epigenomics data to predict gene expression in lung cancer,” BMC Bioinformatics, 2015.

[88] C. Cheng et al., “A statistical framework for modeling gene expression using chromatin features and application to modENCODE datasets,” Genome Biol., 2011.

[89] D. B. Kell, “Metabolomics, machine learning and modelling: Towards an understanding of the language of cells,” in Biochemical Society Transactions, 2005.

[90] K. Märtens, J. Hallin, J. Warringer, G. Liti, and L. Parts, “Predicting quantitative traits from genome and phenome with near perfect accuracy,” Nat. Commun., 2016.

[91] A. L. Swan, A. Mobasheri, D. Allaway, S. Liddell, and J. Bacardit, “Application of Machine Learning to Proteomics Data: Classification and Biomarker Identification in Postgenomics Biology,” Omi. A J. Integr. Biol., 2013.

[92] F. Eduati et al., “Prediction of human population responses to toxic compounds by a collaborative competition,” Nat. Biotechnol., 2015.

[93] C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle, “Deep learning for computational biology.,” Mol. Syst. Biol., vol. 12, no. 7, p. 878, 2016.

[94] A. Jabeen, N. Ahmad, and K. Raza, “Machine learning-based State-of-the-Art methods for the classification of RNA-Seq data,” in Lecture Notes in Computational Vision and Biomechanics, 2018.

[95] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature. 2015.

[96] D. Bahdanau, K. H. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015.

[97] A. Graves, A. R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2013.

[98] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, “Deep learning of the tissue-

regulated splicing code," Bioinformatics, 2014.

[99] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning," Nat. Biotechnol., 2015.

[100] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bull. Math. Biophys., 1943.

[101] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in Journal of Machine Learning Research, 2011.

[102] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, 1986.

[103] J. Wan et al., "A Critical Review of Recurrent Neural Networks for Sequence Learning arXiv : 1506 . 00019v2 [cs . LG] 29 Jun 2015," Int. J. Comput. Vis., 2015.

[104] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in 2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings, 2014.

[105] G. E. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines BT - Neural Networks: Tricks of the Trade," Neural Networks: Tricks of the Trade, 2012.

[106] D. H. Hubel and T. N. Wiesel, "The period of susceptibility to the physiological effects of unilateral eye closure in kittens," J. Physiol., 1970.

[107] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016.

[108] C. Szegedy et al., "Going deeper with convolutions," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015.

[109] J. Zhou and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning-based sequence model," Nat. Methods, 2015.

[110] A. L. Caterini and D. E. Chang, "Recurrent neural networks," in SpringerBriefs in Computer Science, 2018.

[111] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput., 1997.

[112] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014.

[113] J. J. Cannone et al., "The Comparative RNA Web (CRW) Site: An online database of comparative sequence and structure information for ribosomal, intron, and other RNAs," BMC Bioinformatics, 2002.

[114] W. Li and A. Godzik, “Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences,” Bioinformatics, 2006.

[115] Y. Huang, B. Niu, Y. Gao, L. Fu, and W. Li, “CD-HIT Suite: A web server for clustering and comparing biological sequences,” Bioinformatics, 2010.

[116] P. Steffen, B. Voß, M. Rehmsmeier, J. Reeder, and R. Giegerich, “RNAshapes: An integrated RNA analysis package based on abstract shapes,” Bioinformatics, 2006.

CHAPTER 7 APPENDIX

7.1 APPENDIX I: Code for Training the Model

Importing Python Libraries

```
import pickle
from sklearn.model_selection import train_test_split
import keras
import tensorflow as tf
from tensorflow.python.keras.models import Model, Input,Sequential
from tensorflow.python.keras.layers import Masking, GRU,Embedding, Dense, TimeDistributed,Bidirectional,concatenate, Dropout, LSTM
from tensorflow.python.keras.metrics import categorical_accuracy
from tensorflow.python.keras.layers.convolutional import Convolution1D, MaxPooling1D
from tensorflow.python.keras.callbacks import EarlyStopping, ModelCheckpoint,ReduceLROnPlateau
from keras import backend  as K

from keras.preprocessing import text, sequence
from keras.preprocessing.text import Tokenizer
from keras.utils import to_categorical, plot_model
from keras.layers.core import Flatten, Reshape
```

Initializing Training

```
with open ('data_file_2.pkl','rb') as data:
    data_set=pickle.load(data)
records = []
str_data =[]
for i in data_set:
    records.append(i[0])
    str_data.append(i[1])


characters=set()
for record in records:
    for char in record:
        characters.add(char)
print(characters)
print(Len(characters))
count = -1
AA_list = ['U','C','A','G']
aa_index = {char:AA_list.index(char) for char in AA_list}


# The first indices are reserved
aa_index = {k:(v+2) for k,v in aa_index.items()}
aa_index["<PAD>"] = 0
aa_index["<START>"] = 1


reverse_aa_index = dict([(value, key) for (key, value) in aa_index.items()])
def aa_review(text):
    return [aa_index.get(i,'?') for i in text]

seq_data = [aa_review(record) for record in records]

records = 0

tokenizer_decoder = Tokenizer(char_level=True)
tokenizer_decoder.fit_on_texts(str_data)
tokenizer_decoder.word_index = {'b':1,'f':2}
target_data = tokenizer_decoder.texts_to_sequences(str_data)
target_data = sequence.pad_sequences(target_data, maxlen=4381, padding='post')
str_data = to_categorical(target_data)
```

```
train_data, test_data, train_label, test_label = train_test_split(seq_data, str_data, test_size = 0.2, random_state = 0)

seq_data = 0
str_data=0
def decode_aa_review(t):
    return ' '.join([reverse_aa_index.get(i,'?') for i in text])

maxlen = 4381
```

```
train_data = keras.preprocessing.sequence.pad_sequences(train_data,
                                                        value=0,
                                                        padding='post',
                                                        truncating='post',
                                                        maxlen=maxlen)

test_data = keras.preprocessing.sequence.pad_sequences(test_data,
                                                       value=0,
                                                       truncating='post',
                                                       padding='post',
                                                       maxlen=maxlen)
```

Defining Layers and Setting Parameters

```
embedding_dim=64

input = Input(shape=(maxlen,))
x1 = Embedding(input_dim=6, output_dim=64, input_length=maxlen)(input)
conv =  Convolution1D(filters=64,kernel_size=100,
                      padding='same',activation='relu',name='conv1')(x1)

x = Bidirectional(GRU(units=256, return_sequences=True, recurrent_dropout=0.5))(conv)
conca_output = concatenate([conv, x,x1])
x_1 = Dropout(0.5)(conca_output)
y = TimeDistributed(Dense(3, activation="softmax"))(x_1)
model = Model(input, y)
model.summary()
```

Estimating Q2 accuracy, Optimization and Compiling

```
def q2_acc(y_true, y_pred):
    y = tf.argmax(y_true, axis=-1)
    y_ = tf.argmax(y_pred, axis=-1)
    mask = tf.greater(y, 0)
    return K.cast(K.equal(tf.boolean_mask(y, mask), tf.boolean_mask(y_, mask)), K.floatx())

model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy", q2_acc])
model.load_weights('best_model.h5')
es = EarlyStopping(monitor='val_q2_acc', mode='max', patience=3,verbose=1)
mc = ModelCheckpoint('best_model.h5', monitor='val_q2_acc', mode='max', verbose=1, save_best_only=True)
reduce_lr = ReduceLROnPlateau(monitor='val_q2_acc', factor=0.5,mode='max',
                              patience=1, min_lr=0.00001)

history = model.fit(train_data, train_label, batch_size=64, epochs=50,
                    validation_data=(test_data, test_label ), callbacks=[es, mc,reduce_lr],verbose=1)
model.save('model_strand_sep.h5')
```

7.2 APPENDIX II: Code for Predicting Site Accessibility (Final Package)

As published on Code Ocean. DOI: https://doi.org/10.24433/CO.4001375.v1

Importing Python Libraries

```
import pickle
import keras
import tensorflow as tf
from tensorflow.python.keras.models import Model, Input,Sequential
from tensorflow.python.keras.layers import Masking, GRU,Embedding, Dense, TimeDistributed,Bidirectional,concatenate, Dropout, LSTM
from tensorflow.python.keras.layers.convolutional import Convolution1D, MaxPooling1D
from keras import backend  as K
from keras.layers.core import Flatten, Reshape
```

Initializing Model

```
AA_list = ['U','C','A','G']
aa_index = {char:AA_list.index(char) for char in AA_list}

# The first indices are reserved
aa_index = {k:(v+2) for k,v in aa_index.items()}
aa_index["<PAD>"] = 0
aa_index["<START>"] = 1


reverse_aa_index = dict([(value, key) for (key, value) in aa_index.items()])
def aa_review(text):
    return [aa_index.get(i,'?') for i in text]


def decode_aa_review(t):
    return ' '.join([reverse_aa_index.get(i,'?') for i in text])

maxlen = 4381
```

Defining Layers and Setting Parameters

```
Input = Input(shape=(maxlen,))
x1 = Embedding(input_dim=6, output_dim=64, input_length=maxlen)(Input)
conv =  Convolution1D(filters=64,kernel_size=100,
                      padding='same',activation='relu',name='conv1')(x1)


x = Bidirectional(GRU(units=256, return_sequences=True, recurrent_dropout=0.5))(conv)
conca_output = concatenate([conv, x,x1])
x_1 = Dropout(0.5)(conca_output)
y = TimeDistributed(Dense(3, activation="softmax"))(x_1)
model = Model(Input, y)
```

One-Hot Encoding for Output

```
import numpy as np
def onehot_to_seq(oh_seq, index,length):
    s = ''
    oh_seq = oh_seq.reshape(4381,3)
    prob = oh_seq[:length,2]
    #class_p = oh_seq[:length,1:]
    for i in range(length):
        i = np.argmax(oh_seq[i])

        if i != 0:
            s += index[i]
        else:
            break
    return s,prob
```

Site Accessibility Prediction

```
72  def seq_predict(seq_1):
73
74
75      revsere_decoder_index = {1:'b',2:'f'}#{value:key for key,value in {'b':1,'f':2}}
76      revsere_encoder_index = dict([(value, key) for (key, value) in aa_index.items()])
77      seq_2 = aa_review(seq_1)
78      seq_2=seq_2+[0 for i in range(4381-Len(seq_2))]
79      seq_2=np.array([seq_2])
80      y_train_pred = model.predict(seq_2)
81      return onehot_to_seq(y_train_pred,revsere_decoder_index,Len(seq_1))
82
83
84  seq='AGGAAAGUCCCGCCUCCAGAUCAAGGGAAGUCCCGCGAGGGACAAGGGUAGUACCCUUGGCAACUGCACAGAAAACUUACCCCUAAAUAUUCAAUGAGGAUUUGAUUCGACUCUUACCUUGGCGACA
85  y_pred,proba_free = seq_predict(seq)
86
87  proba_free = proba_free.tolist()
88
89  with open('../results/result.txt','w+') as f:
90      f.write(seq+'\n'+'\n'+'\n')
91      f.write(y_pred.upper()+'\n'+'\n'+'\n')
92      for i in proba_free:
```

Note: User can define sequence in line 84 (seq = ‘ ’) to estimate site accessibility.

LIST OF PUBLICATIONS

Journal Paper

- Shubham Mittal, Rajkumar Chakraborty, Yasha Hasija. (2020) **Predicting RNA Site Accessibility using Deep Learning.** (Springer Journal). [Communicated]

Conference Papers

- Shubham Mittal, Yasha Hasija. **Studying Network features in Systems Biology using Machine Learning.** Information and Communication Technology for Intelligent Systems (ICTIS 2020), May 15-16, 2020 (Springer). [Accepted]

- Shubham Mittal, Yasha Hasija. **RNA Secondary Structure Prediction using Machine Learning: A Review.** International Conference on Computing, Communication and Automation (ICCCA 2020), October 30-31, 2020 (IEEE). [Accepted]

- Shubham Mittal, Yasha Hasija. **Advancements in RNA Secondary Structure Prediction using Machine Learning Methods.** International Conference for Innovation in Technology (INOCON 2020), November 6-8, 2020 (IEEE) [Accepted].

Book Chapter

- Shubham Mittal, Yasha Hasija (2020) **Applications of Deep Learning in Healthcare and Biomedicine.** In: Dash S., Acharya B., Mittal M., Abraham A., Kelemen A. (eds) Deep Learning Techniques for Biomedical and Health Informatics. Studies in Big Data, vol 68. Springer. https://doi.org/10.1007/978-3-030-33966-1_4 [Published]

Source Code

- Rajkumar Chakraborty, Shubham Mittal, Yasha Hasija (2020) **Predicting RNA Site Accessibility using Deep Learning** [Source Code]. https://doi.org/10.24433/CO.4001375.v1 [Published]