RNA Secondary Structure Prediction Using Machine Learning

Vinayak Singh Bhadoriya

19th April 2024

MANCHESTER
1824

The University of Manchester

Contents

Declaration **3**

List of Figures **5**

List of Tables **6**

List of Abbreviations **7**

Abstract **8**

1 Introduction **9**
- 1.1 Context and Motivation 9
- 1.2 Aims and Objectives 10
- 1.3 Report Structure 12

2 Background **14**
- 2.1 RNA Structure and Function 14
  - 2.1.1 Wobble Base Pairing 15
  - 2.1.2 The bpseq and dot bracket notation 15
- 2.2 Structural motifs in secondary structures 16
- 2.3 RNA Secondary Structure Prediction Algorithms and Tools . . 21
- 2.4 Machine Learning 24
  - 2.4.1 Stochastic Gradient Descent 24
  - 2.4.2 Support Vector Machines 25

3 Design Methodology and Implementation **26**
- 3.1 Initial Experimentation 26
- 3.2 Data Collection 27
- 3.3 Feature Extraction 27

3.3.1 Base Pairing 27
3.3.2 Dangling Ends 28
3.3.3 Hairpin Loop 28
3.3.4 Stacks 28
3.3.5 Bulges 29
3.3.6 Internal Loops 29
3.3.7 Bifurcation Loops 29
3.4 Implementation and Training 31
3.4.1 Preliminaries 31
3.4.2 Learning Algorithm 32
3.4.3 Loss-Augmented Inference 32
3.4.4 Training 34

4 Results **35**
4.1 Evaluation Metrics 35
4.2 Results 36
4.3 Discussion 37

5 Conclusion **39**
5.1 Summary 39
5.2 Future Work 39

A Appendix **47**
A.1 Secondary Structures 47
A.1.1 Bacteria, *Anabaena variabilis* 47
A.1.2 Archaea, *Haloquadratum walsbyi* 47
A.1.3 Bacteria, *Microbacterium* 47

Declaration

I declare that this report is my own work and that all sources of information have been acknowledged and referenced. No portion of the work referred to in this report has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Acknowledgements

I would like to thank my supervisor, Dr. Konstantin Korovin, for his guidance and support throughout the project. He has been a constant source of motivation and has provided me with valuable insights and feedback that have helped me in completing this project.

I would also like to thank my family and friends for their encouragement and support, throughout the course of this project. Their words and motivation helped me every day.

List of Figures

1.1 RNA folding process from Zhao et al.(2021)[12] 10
1.2 Primary structure of RNA 11

2.1 Secondary Structure in Shell layout of Anabaena variabilis . . 17
2.2 Secondary Structure in Kamada-Kawai layout [18] of Anabaena variabilis 17
2.3 Hairpin Loop and Stacking Regions 18
2.4 Secondary Structure visualisation of the bacteria using ViennaRNA [23] 19
2.5 Secondary Structure of Archaea *Haloquadratum walsbyi* using ViennaRNA [23] 20
2.6 Secondary Structure of a microbacterium using ViennaRNA [23] 22

List of Tables

2.1 RNA Bases and Complementary Watson-Crick Base Pairings . 15

4.1 Evaluation of our model on the TestSetA dataset. 36
4.2 Accuracy for each scoring model: The thermodynamic model (TM), the machine learning-based model (ML) trained with TrainSetA, and the integrated model (TM+ML). By Akiyama et al. 37
4.3 Performance metrics for various RNA secondary structure prediction methods. PPV: Positive Predictive Value, SEN: Sensitivity, F: F1-score. Taken from Sato et al.[46] 38

List of Abbreviations

SVM	Support Vector Machine
SSVM	Structured Support Vector Machine
RNA	Ribonucleic Acid
ncRNA	Non-coding RNA
mRNA	Messenger RNA
rRNA	Ribosomal RNA
tRNA	Transfer RNA
DNA	Deoxyribonucleic Acid
A	Adenine
G	Guanine
U	Uracil
C	Cytosine
bpseq	Base Pair Sequence
DB	Dot Bracket
NMR	Nuclear Magnetic Resonance
CRF	Conditional Random Field
CLLM	Conditional Log Linear Model
SCFG	Stochastic Context Free Grammar
NNDB	Nearest Neighbour Database
SEN	Sensitivity
PPV	Positive Predictive Value

Abstract

Non-coding RNAs (ncRNAs) play crucial roles in various biological processes, including gene regulation, cellular signaling, and disease mechanisms. Understanding these functions requires accurate prediction of RNA secondary structures. This report focuses on the exploration and advancement of RNA (Ribonucleic Acid) structure prediction methodologies and explores various computational approaches, including thermodynamic models, machine learning-based models, and deep learning techniques. I propose a machine learning approach to predict RNA secondary structures using the structured support vector machine (SSVM) model first put forward as a solution by Akiyama et al. The model's performance was evaluated using sensitivity, specificity, and F-value metrics, which are more appropriate for this context than traditional accuracy metrics. Furthermore, this work emphasizes the significance of rich parameterization in improving RNA structure predictions and explores the integration of thermodynamics with deep learning approaches and underscores the importance of dataset diversity in achieving robust and generalizable predictions. Future work includes incorporating pseudoknots, exploring larger datasets, and investigating faster computation methods to improve model efficiency.

Chapter 1

Introduction

1.1 Context and Motivation

Ribonucleic acid (RNA) is a fundamental polymeric molecule, essential for all known forms of life playing critical roles in various biological processes such as the regulation of gene expression and protein synthesis. It is made up of nucleotides, which are ribose sugars attached to nitrogenous bases and phosphate groups-the nitrogenous bases include adenine, guanine, uracil, and cytosine(denoted by the letters A, G, U, and C respectively) and the RNA molecule itself can have a variety of lengths, structures, and substructures.[1]

The chemical structure of RNA is similar to its more famous counterpart Deoxyribonucleic acid(DNA), however RNA mostly exists in the single-stranded form, contains ribose sugars instead of deoxyribose sugars, and has Uracil(an unmethylated form of Thymine [2]) instead of Thymine as the complementary base to Adenine.

Non-coding RNAs(ncRNA) are functional RNA molecules that are transcribed from DNA but not translated into proteins-these are linked to various pivotal functions in human biology and disease. Many ncRNAs like messenger RNA(mRNA), ribosomal RNA(rRNA), and transfer RNA(tRNA) which were earlier thought to play minor roles in biological processes have now been found to perform crucial functions; ribosomal protiens provide the scaffold allowing rRNAs to catalyze peptide bond formation[3]; and tRNAs are the “adaptor molecules” that allow the translation of mRNA into proteins[4], tRNA derived fragments also play a role in gene expression regulation as regulatory RNAs[5]. Efforts have even been made to use ncRNAs for potential cancer therapy applications[6].

The reason to highlight the above examples is to underline the importance of RNA in biological processes and the need to understand its function. To that end, it has also been shown that the secondary structure of RNA provides the scaffold necessary for the tertiary structure formation[1][7], and conformations and structures of ncRNAs are highly related to their stability and provide an insight into their functions and mechanisms of action [8]: for example, stem-loop substructures in mRNA bind to proteins and regulate its synthesis[9]; Palmenber and Sgro[10] have even highlighted the regions of picornviral RNA secondary structures that are likely to play a significant role in virus biology. And so, secondary structure prediction of RNA is an important problem in bioinformatics and computational biology.

But what exactly is the secondary structure of RNA and how is it formed? As mentioned before, unlike DNA, RNA is mostly single-stranded and in 1960 Fresco et al.[11] first showed that single-stranded RNA can fold onto itself to form a stable secondary structure composed of different substructures which are held together by hydrogen bonds between base pairs. This forms the basis of RNA secondary structure prediction: given a primary sequence of RNA, predict the most likely secondary structure that it will fold into.

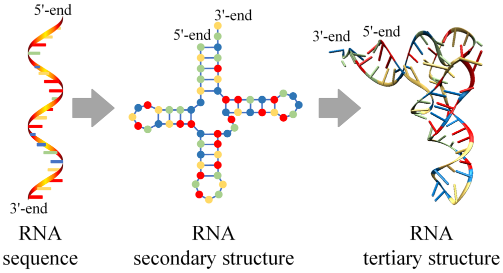

Figure 1.1: RNA folding process from Zhao et al.(2021)[12]

1.2 Aims and Objectives

As postulated by Tinoco and Bustamante[1]:

> "In an RNA, ... the information in the sequence flows linearly, and largely in one direction, first to the secondary and then to the tertiary structure. An RNA molecule can thus be thought of as

> possessing a hierarchical structure in which the primary sequence determines the secondary structure..."

This presents the groundwork for a prediction algorithm-the primary structure is a linear sequence of nucleotides linked together by a "backbone" of phosphodiester bonds(1.2) and this structure will then fold onto itself, complementary bases pairing up to form different substructures like hairpins, bulges, and internal loops-structural motifs which will formally be defined and discussed in more detail in the next chapter-ultimately forming the secondary structure of the RNA molecule.

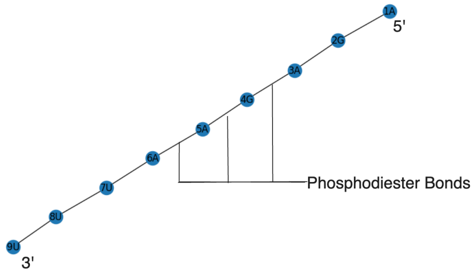

Figure 1.2: Primary structure of RNA

This project aims to predict that exact process: develop a machine learning model that can predict the most probable secondary structure of an RNA molecule given its primary sequence, correctly identifying the base pairings that will take place and the substructures that will form. In doing so, I also hope to outperform the current state-of-the-art algorithms and tools in the field of RNA secondary structure prediction in terms of accuracy and efficiency, a much-needed improvement given that the complexity of dynamic programming for secondary structure prediction is to the order of $O(n^3)$ [13].

The machine learning architecture that I will be employing for this task is a Support Vector Machine(SVM) model, a supervised learning algorithm first introduced by Cortes and Vapnik[14]. A modified variation, SSVM, was used by Akiyama et al.[15] in 2018 to propose a solution that integrated the thermodynamic approach to RNA secondary structure prediction with a machine learning approach, achieving the highest accuracy at the time.

My project builds on the work of Akiyama et al.[15], addressing limitations in the proposed solution and the paper, namely detailing how to use the dynamic programming approach to RNA secondary structure prediction in conjunction with the SVM model, and how to improve the feature extraction process to better represent the primary sequence of RNA.

The objectives of this project are as follows:

- To collect a dataset of RNA sequences and their secondary structures.
- To extract features from the primary sequences of RNA.
- To develop a Support Vector Machine model for RNA secondary structure prediction.
- To train the model on the dataset and evaluate its performance.

On the point of evaluation, the performance metrics that are generally used to evaluate the accuracy of prediction are Sensitivity, Specificity, Positive Predictive Value. These metrics will be used to evaluate the performance of the SVM model in this project and will be discussed in more detail in the penultimate chapter on results.

1.3 Report Structure

The report consists of 5 chapters and an appendix, structured as follows:

- **Chapter 1: Introduction** - This chapter provides an overview of the project, putting it into a relevant context and the motivation behind it, the aims and objectives, and the structure of the report.
- **Chapter 2: Background** - This chapter provides the necessary background information on RNA structure and function, a discussion of the state-of-the-art RNA secondary structure prediction algorithms and tools, structural motifs in RNA secondary structures, and finally word embeddings, and support vector machines-the last two providing the necessary background for the machine learning model used in this project.
- **Chapter 3: Design Methodology and Implementation** - This chapter details the data collection process, feature extraction, the SVM model, the learning algorithm, and the implementation and training process.

- **Chapter 4: Results** - This chapter details the evaluation metrics used, the results obtained, and a discussion of the results.

- **Chapter 5: Conclusion** - This chapter provides a summary of the project and its findings and whether the project aims and objectives were met, along with suggestions for future work that may take this solution further.

- **Appendix** - This chapter provides additional information on the project, including code snippets, and details about the parameters and datasets used.

Chapter 2

Background

This chapter provides the necessary background information required to set the project in relevant context based on the scientific literature in the field of RNA secondary structure prediction, beginning with a discussion of RNA structure, followed by a discussion of structural motifs in RNA secondary structures and then a deeper dive into secondary structure prediction algorithms and tools detailing the approaches and strategies adopted by them, and finally an introduction to concepts which will form the foundation of the machine learning model used in this project-word embeddings and support vector machines.

2.1 RNA Structure and Function

I mentioned before that RNA is a polymeric molecule made up of nucleotides and primary structures of RNA are linear sequences of nucleotides linked together by phosphodiester bonds, these nucleotides in turn are each made up of a ribose sugar, a nitrogenous base, and a phosphate group.

Conventionally, the four nitrogenous bases-Adenine, Guanine, Uracil, and Cytosine-are denoted by the letters A, G, U, and C respectively, and the primary sequence is denoted by writing the letters of the bases in order from the 5' end to the 3' end of the RNA molecule. The 5' end of the RNA molecule is the end where the phosphate group is attached to the 5' carbon of the ribose sugar, while the 3' end is where the hydroxyl group is attached to the 3' carbon of the ribose sugar. [16]

Adenine and Guanine have a double-ring structure and are known as

Base	Symbol	Complementary Base
Adenine	A	Uracil
Guanine	G	Cytosine
Uracil	U	Adenine
Cytosine	C	Guanine

Table 2.1: RNA Bases and Complementary Watson-Crick Base Pairings

purines, while Cytosine and Uracil have a single-ring structure and are known as pyrimidines. Their symbols and complementary base pairings are shown in Table 2.1.

The secondary structure consists of one single polynucleotide chain, with base pairing occurring when folding takes place between complementary bases.

There is a caveat, however, that is hinted at in the table 2.1 above. The base pairings shown in the table are the "canonical" Watson-Crick base pairings, but RNA can also form non-canonical base pairings where the bases are not strictly complementary to each other.

2.1.1 Wobble Base Pairing

Simply put, the canonical Watson-Crick pairings are the base pairings that can form between the bases of RNA where the bases are strictly complementary to each other, while wobble base pairings do not follow this general rule.

In 1966, Crick[17] postulated the "wobble hypothesis" - that the 5' base of the anticodon in tRNA can form non-standard base pairings with the 3' base of the codon in mRNA. This non-standard base pairing is known as "wobble base pairing". This wobble pairing would not be implemented in my project, but it is important in the context of RNA secondary structure prediction as it allows for more flexibility in the base pairings that can form between the bases of RNA.

2.1.2 The bpseq and dot bracket notation

Representing the primary structure was fairly straightforward, but portraying the secondary structure is still a debated problem with many different notations and representations being used. Two of the most common representations are the bpseq and dot bracket notation, which I will be employing in this project as well. The secondary structure contains the pairs formed between the bases of RNA-the nucleotides that form the linear sequence. How can this fact be reflected in the representation? The bpseq notation solves this problem by indexing. Take for example this short RNA sequence in bpseq notation:

```
1 A 5
2 A 0
3 A 0
3 G 0
4 C 0
5 U 1
6 G 0
```

The first column represents the index of the nucleotide in the sequence, the second column represents the base itself, and the third column represents the index of the nucleotide that it is paired with, and if the nucleotide is unpaired, it is denoted by 0.

The dot bracket notation is a more compact representation, which makes use of the fact that base pairing is a binary relationship creatively. The same RNA sequence in dot bracket notation would look like this:

```
AAAGCUG
(....).
```

The primary sequence is written on the first line, while the secondary structure is written on the second line denoted by "." for unpaired bases and "()" for the two paired bases. The base which appears first in the sequence is denoted by the opening parenthesis, and the base which appears later is denoted by the closing parenthesis, and so all the opening parentheses are closed in the correct order by the closing parentheses.

2.2 Structural motifs in secondary structures

We have already talked about the fact that the primary sequence of RNA folds onto itself to form secondary structures, but it also important to understand the different patterns and substructures that are to be seen in these secondary structures, as they form the basis of the prediction algorithms and tools that are used to predict them.

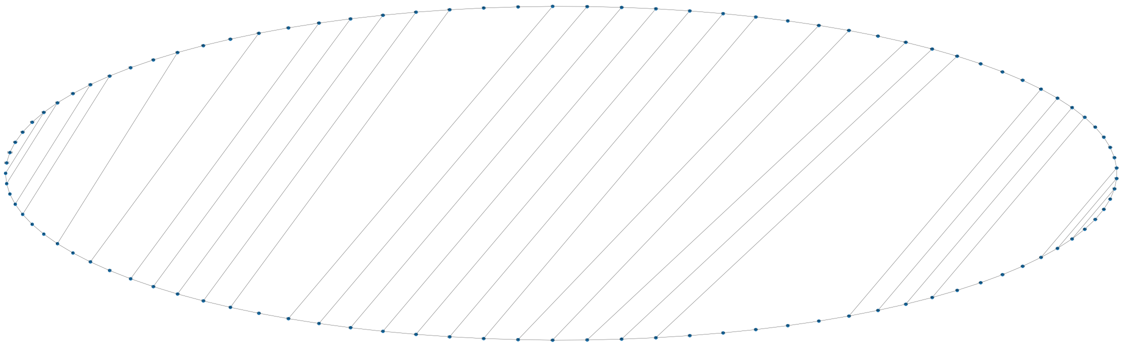

Figure 2.1: Secondary Structure in Shell layout of Anabaena variabilis

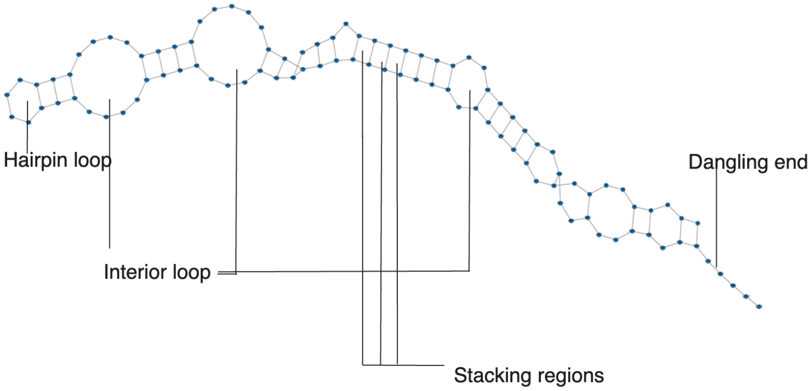

Figure 2.2: Secondary Structure in Kamada-Kawai layout [18] of Anabaena variabilis

Figures 2.1 and 2.2 show the secondary structure of a bacteria of the species *Anabaena variabilis* taken from the bpRNA database [19], that we will be using to explore a few motifs. These were generated using the networkx library in Python [20]. and the sequence and the dot bracket notation for this structure can be found in the appendix A.1.1. By convention, the sequence is numbered from the 5' end, denoting by iN the nucleotide N at index i in

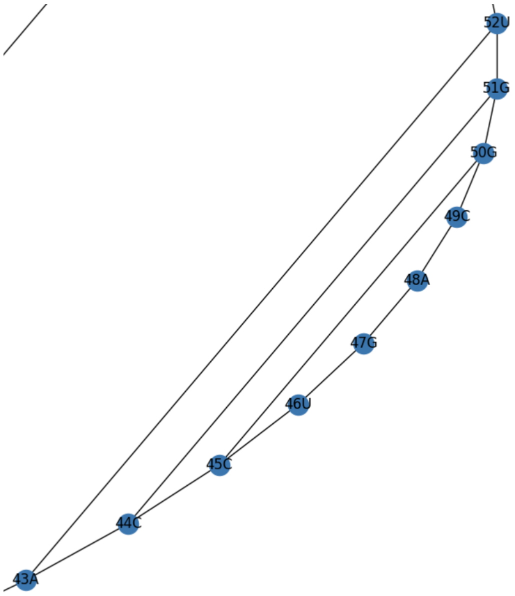

Figure 2.3: Hairpin Loop and Stacking Regions

the sequence for $1 \leq i \leq n$ where n is the length of the sequence. The entire sequence is denoted by S and the $S_i j$ denotes the subsequence from iN to jN, where $1 \leq i \leq j \leq n$.

In figure 2.1, the nucleotides form the vertices of the shell, and the arcs formed by those vertices represents the phosphodiester bonds between the nucleotides. These arcs are referred to as the **exterior edges** of the shell. The line segments that connect the vertices are referred to as the **interior edges** of the shell, and they represent the base pairs formed between the nucleotides.

An entire structure is valid if the line segments or “chords” do not touch each other. Not allowing the chords to touch is a constraint that allows only one base pair to form between two nucleotides,

Faces in the shell - Any region completely enclosed by edges is referred to as a face and each of these regions can be categorised as a structural motif. All the motifs are defined in relation to the shell or “semi-circular” layout of the secondary structure, however, I have included other layouts as well to provide a more comprehensive view of the structure.

In Figure 2.3, take, for instance, the region from 45C to 50G, bounded by a single interior edge and 5 exterior edges - this is a **hairpin** or a **stem**

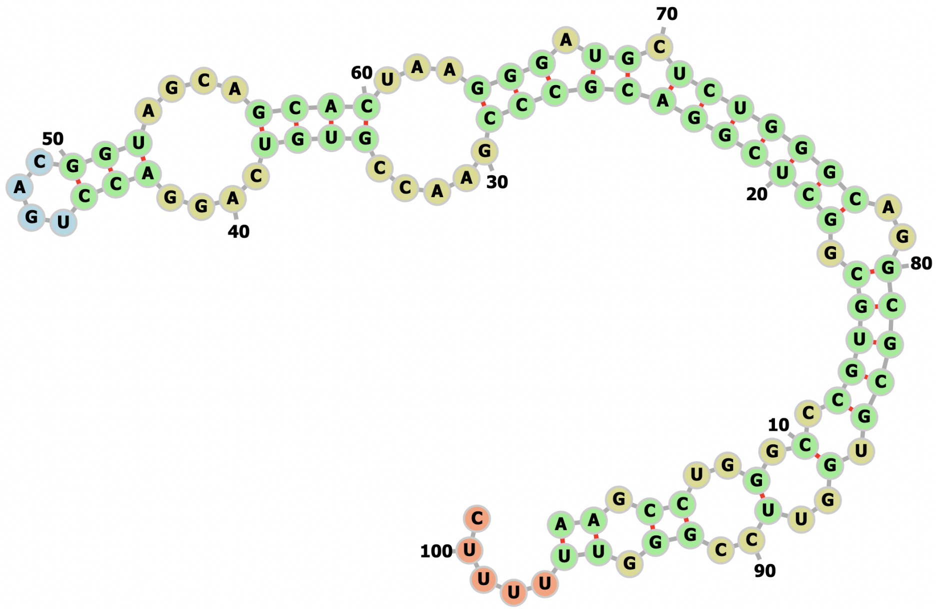

Figure 2.4: Secondary Structure visualisation of the bacteria using ViennaRNA [23]

loop and so is any face bounded by a single interior edge. The same can also be seen in the far left of Figure 2.2.

What about the area bounded by 43A 44C and 51G 52U, as well as the one bounded by 44C 45C and 50G 51G? Any face bounded by two interior edges with the two interior edges being separated by a single exterior edge on both sides is referred to as a **stacking region** or simply a **stack**. These are an integral part of the structure since it has been shown that more than the hydrogen bond face interaction, it is these "stacks" that provide stability to the molecule [21] [22].

Sometimes the face is bounded by two interior edges with the two interior edges being separated on one side by a single exterior edge and on the other side by two or more exterior edges - this is referred to as a **bulge**, seen more clearly in the same secondary structure created using the *forna* tool available in the ViennaRNA package in Figure 2.4 [23] at nucleotide 70C.

What if the face is bounded by two interior edges with the two interior edges being separated by two or more exterior edges on *both* sides? This is referred to as an **internal loop**, one of the many in this particular structure displayed in the Kamada-Kawai layout in Figure 2.2 from 38U to 57G.

One particular motif that does not rely on faces or bounded regions is a **dangling end**-a single-stranded region that is not paired with any other nucleotide on one side. It can be seen in Figure 2.2 from nucleotide 96U to 101C.

A particular motif that is not seen in this structure is a **bifurcation loop** or a **multi-loop**. It is defined as a face which is bounded by three or more interior edges-seen clearly in the secondary structure of the Archaea *Haloquadratum walsbyi* in Figure 2.5. The sequence and the dot bracket notation for this structure can be found in the appendix A.1.2.

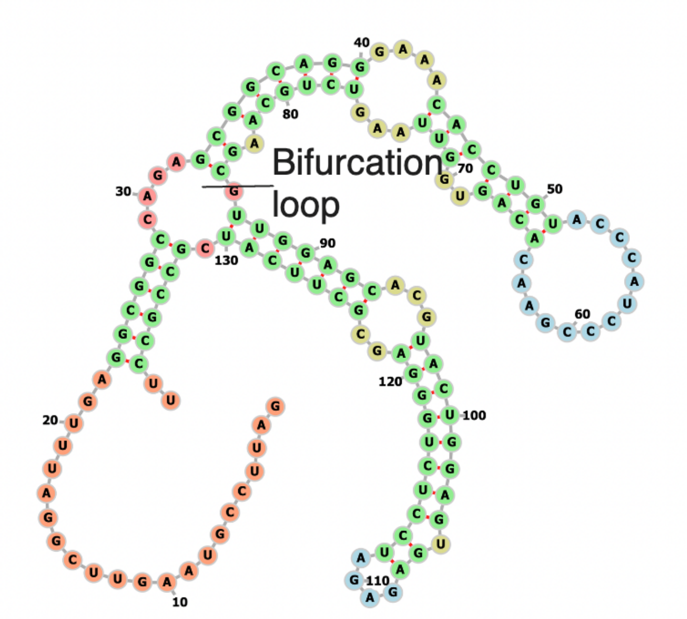

Figure 2.5: Secondary Structure of Archaea *Haloquadratum walsbyi* using ViennaRNA [23]

It is also worth talking about **pseudoknots** that though not as common as the other motifs do appear in a few cases-I will not be accounting for this motif in this project, however, so as to lower the computational complexity of the prediction algorithm and speed it up, since the complexity of the prediction algorithm with pseudoknots is to the order of $O(n^6)$ [24].

According to Staple and Butcher[25]:

> "First recognized in the turnip yellow mosaic virus[26], a pseu-

> doknot is an RNA structure that is minimally composed of two helical segments connected by single-stranded regions or loops"

Put in simpler terms, an unpaired nucleotide in a loop can base pair with a nucleotide outside the loop in a single stranded region, forming a pseudoknot. In terms of the shell layout, this would mean interior edges or chords intersecting each other.

Seen in Figure 2.6 in the secondary structure of a microbacterium. Sequence and the dot bracket notation for this structure can be found in the appendix A.1.3.

2.3 RNA Secondary Structure Prediction Algorithms and Tools

Experimental approaches, such as X-ray crystallography[27] and NMR spectroscopy[28], are by far the most accurate way to determine the secondary structure of RNA, however because of the high cost and the time-consuming nature of these methods, computational methods have been sought after to predict RNA secondary structures.

In 1978, Nussinov et al.[29] first proposed a simple but powerful dynamic programming algorithm to predict RNA secondary structures, based on some simplifying assumptions: the weight of A-U and G-C base pairs is the same, the nucleotide will not pair with an adjacent nucleotide, and "no two matches will cross each other when drawn in the interior of the loop" - similar to our constraint of not allowing chords to intersect-and this resulted in the maximum number of base pairs being formed. Though the assumption that the stability of the structure depends on the bonds formed between the bases and not on the various substructures that those bases form is not exactly correct, this was a seminal paper that led the groundwork for future algorithms.

Formalising the constraints and assumptions, the algorithm can be described as follows:

Input: A sequence S of length n.

Output: A secondary structure π_S of S represented as a matching of indices with the following constraints:

- Possible matches: AU(or UA) or GC(or CG).
- No two matches cross each other.

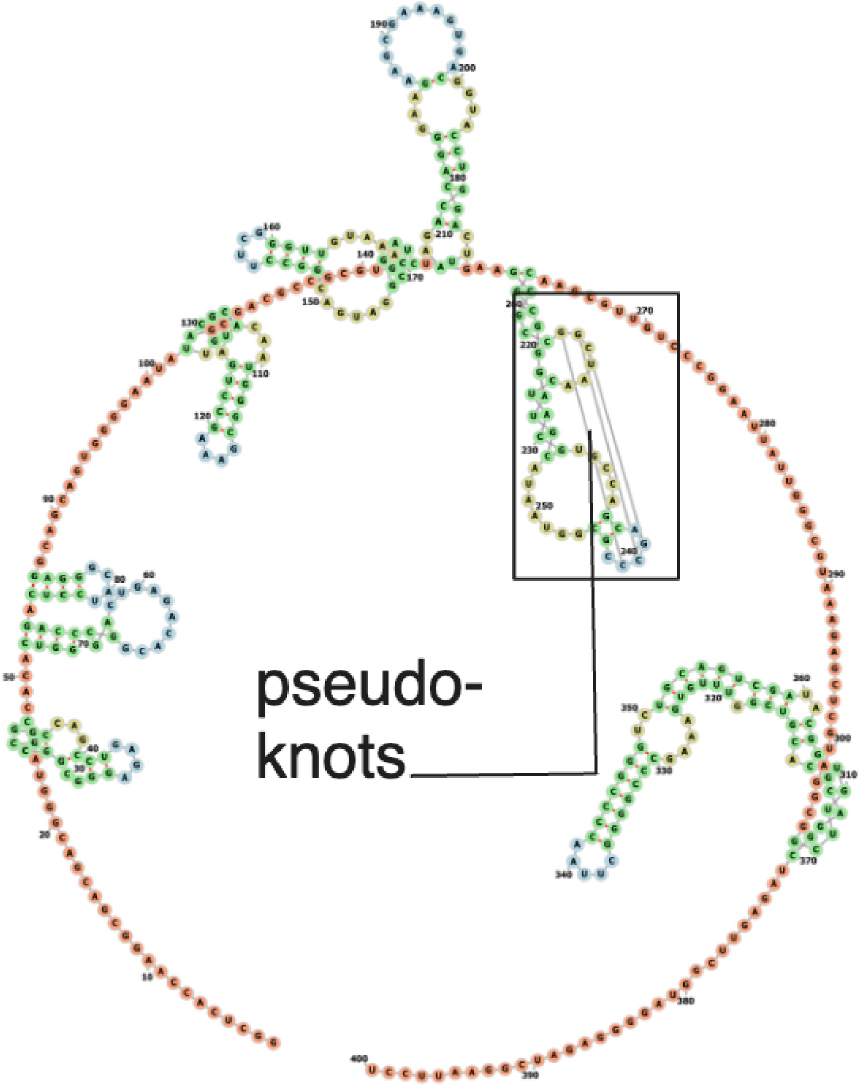

Figure 2.6: Secondary Structure of a microbacterium using ViennaRNA [23]

- A match (i, j) is allowed only if $j - i > 1$.
- Matching is one-to-one.

Defining the Matrix $M(i, j)$ as the maximum number of base pairs that can be formed in the subsequence S_{ij},

$$M_{ij} = \max \{|N||N \text{ is a matching of } S_{ij} \text{ satisfying the constraints}\} \tag{2.1}$$

The algorithm can be described as follows:
Initialize the matrix M with zeros.

$$M_{ij} = \max \begin{cases} M_{i,j-1} \\ \max_{i \leq k < j}\{M_{i,k-1} + M_{k+1,j-1} + 1\} \end{cases} \tag{2.2}$$

Methods, before Nussinov's algorithm, were proposed that used a thermodynamic approach to predict RNA secondary structures, this approach used thermodynamic parameters to "score" features like motifs and basepairs that appeared in a structure and then used a folding algorithm to sum up the free energies of each feature or substructure and then find the structure with the lowest or "minimum free energy". These thermodynamic parameters were derived from careful experiments and estimation [22][30][31], and therein lies the limitation of this approach-the calorimetric measurements of RNA structures used to derive these parameters are extremely laborious and time-consuming.

Tinoco et al.[21] devised a basic method to estimate secondary structures, and in 1975 Pipas and McMahon[32] proposed a method to predict the secondary structure using a thermodynamic approach, but the latter computed in $O(2^N)$ time, which was not feasible for long sequences.

It was only in 1981 that Zuker and Stiegler[13] came up with a fusion of the dynamic programming algorithm and the thermodynamic approach, and presented a method that found the minimum free energy secondary structure efficiently using the published values of the time. This folding algorithm is still used today in popular RNA secondary structure prediction tools and methods like MFold[33], UNAfold[34], and ViennaRNA[23].

Since this approach places an emphasis on the thermodynamic values of the nucleotides with the closest neighbours, this is referred to as the "nearest neighbour" approach as well and in 2009, Turner and Mathews[35] published a database of these thermodynamic parameters titling it the "Nearest Neighbour Database(NNDB)". The database aims to assemble and curate all the thermodynamic parameters for RNA folding and provides documentation and tutorials on how to use them.

This is also the folding algorithm that I will be using in this project in my model, albeit with a different approach.

In a **machine learning** or **weighted** approach, the thermodynamic scores are replaced by weights that are either guessed or, more commonly, learned from a dataset of known secondary structures. Both these approaches are also referred to as "discriminative" methods, which are contrasted with "generative" methods which use **probabilistic models** to predict the secondary structure. A whole host of machine learning models have been proposed in the last two decades, beginning with the work of Do et al.[36] which used Conditional Random Fields(CRFs) and Conditional Log Linear Models(CLLMs) to predict RNA secondary structures, to ensemble models in SPOT-RNA [37], and including the SSVM model proposed by Akiyama et al.[15]-mxfold.

Since I will not be using a probabilistic model in this project, I will not be discussing that approach in a lot of detail but it should be noted that in this approach, the model uses the probabilities of producing certain base pairs and substructures. Like the machine learning approach, the probabilities are either guessed or learned from known structures. A probabilistic Stochastic Context Free Grammar(SCFG) model was first suggested by Durbin et al.[38] in 1998. Elena Rivas, one of the most prominent researchers in the field of RNA secondary structure prediction, even proposed TORNADO[39], "a general purpose parser to produce grammars of single-sequence RNA secondary structure."

2.4 Machine Learning

2.4.1 Stochastic Gradient Descent

Machine learning models learn from data or experience and improve their future predictions based on the data they have seen. This learning is done by adjusting the weights of the model based on the error or "loss" that the model makes in its predictions. Depending on the the error or the loss in the prediction, the weights are adjusted in the direction that minimizes the loss, in order to incorporate the learning. And a way to change weights is through the use of an optimization algorithm, one of which is the Stochastic Gradient Descent(SGD) algorithm.

2.4.2 Support Vector Machines

Support Vector Machines(SVMs) are a class of supervised learning algorithms that can be used for both classification and regression tasks. The algorithm works by finding the hyperplane that best separates the data into two classes, and then uses this hyperplane to make predictions on new data. The hyperplane is chosen in such a way that it maximizes the margin between the two classes, and this is why SVMs are also known as "maximum margin classifiers".

In 2005, Tsochantaridis et al.[40] proposed a novel approach to extend the concept of margin maximization to deal with complex output spaces and proposed an efficient algorithm to solve the resulting optimization problem.

This Structured SVM(SSVM) model was used by Akiyama et al.[15] in 2018 to with L_1 regularization to avoid overfitting to predict the secondary structures

Chapter 3

Design Methodology and Implementation

3.1 Initial Experimentation

Before diving into the SVM model, I wanted to experiment with the concept of using word embeddings as a way to represent the primary sequence of RNA, and in doing so try to somehow “translate” that into its secondary structure.

It is easy to analogise the RNA sequence in terms of a natural language sentence, with the sequences being “sentences”, the nucleotides and substructures forming the “words” of that sentence.

I used the *gensim* library in Python to train a Word2Vec model on the primary sequences of RNA, and then used the embeddings to train a simple neural network to predict the secondary structure. Since the results were not promising and the model was not able to learn the patterns in the data, I decided to move on to other experiments. The reason why the model did not perform well could be because the patterns in the data are not as easily discernible and analogised as in natural language, since the motifs and substructures are more similar to “patterns” than words.

In all the experiments I ran, the outputs produced were absurd and did not make sense. The model was able to learn the fact that A is paired with U and G is paired with C very often, but the model was not able to learn the more complex patterns that are present in the data.

Moving on from this, I decided to use the SVM model, and follow a more conventional setup to create a Secondary Structure Prediction model, as described by Rivas et al.[41].

3.2 Data Collection

While working on the parameter estimation, Andronescu et al.[42] collected a large dataset of about 3000 RNA secondary structures from different sources. Since then, the data sets by Andronescu et al. have been known under different names [42][43], and had widely been employed in the field. However, it did suffer from a shortcoming-as reported by Rivas[41], even though the dataset contained a larger number of sequences, it:

> “...covers only six different RNA structures: small and large subunit rRNAs, tRNAs, tmRNAs, ribonuclease P RNA and signal recognition particle RNAs.”

and Rivas et al.[39] recognising the need to test and train new models on datasets that contained sequences structurally dissimilar came up with different training and test sets. The new datasets include “TrainSetA/TestSetA” and “TrainSetB/TestSetB”, both of which are in the bpseq notation. The former contains the Andronescu data set along with new sequences collected by Lu et al.[44], and that is the set being used in this project.

3.3 Feature Extraction

Using the motifs and substructures discussed in the previous chapter, the features can easily be extracted from the secondary structures in our dataset, and form the feature vector for each sequence to train our model. Taking a look at how some of the features can be extracted:

3.3.1 Base Pairing

The most basic feature that can be extracted is the base pairing itself. The data set is stored in the bpseq notation, and as mentioned before, the bpseq notation contains the indices of the nucleotides that are paired, which means that the AU or GC pairs can be extracted directly from the bpseq notation.

3.3.2 Dangling Ends

Dangling ends are unpaired nucleotides that are not paired with any other nucleotide on one side. They are a feature that can also easily be extracted using the DB notation. In the DB notation, the dangling ends can be seen as a string of dots or unpaired bases right at the beginning or end of the structure, for example, in this toy example:

```
AAAAGCGCU
...(....)
```

The dots at the beginning represent the dangling end at the 5' end of the sequence.

Following is an algorithm to convert the bpseq notation to the dot bracket notation:

```
def bpseq_to_dot_bracket(bpseq_file: list[list]) -> str:
dbnotation = ['.' for i in range(len(bpseq_file))]
for base in bpseq_file:
    if int(base[2]) != 0:
        dbnotation[base[0]-1] = ')'
        dbnotation[base[2]-1] = '('
return ''.join(dbnotation)
```

3.3.3 Hairpin Loop

In terms of the shell layout, a hairpin loop is a face bounded by a single interior edge. In DB notation, the way this would be represented is:

```
(<any number of unpaired bases>)
```

And all the patterns of this form can be extracted from the notation using regular expressions:

```
re.findall(r'\(\.*\)', db_sequence)
```

3.3.4 Stacks

A stack is a face bounded by two interior edges with the two interior edges being separated by a single exterior edge on both sides.

For stacks and the next few substructures, it is important to look at the bpseq notation-iterating over the bases, if a base is paired, we move forward to look for the next paired base. If it so happens that the distance between the first paired base, say i, and the next paired base, say j, is 1, and if the distance between the base paired with i, say k, and the base paired with j, say l,is 1, then we have a stack.

j - i = 1 and k - l = 1.

3.3.5 Bulges

A bulge is a face bounded by two interior edges with the two interior edges being separated on one side by a single exterior edge and on the other side by two or more exterior edges. They are extracted in a similar way to stacks.

Iterate over the bases, if a base is paired, move forward to look for the next paired base. If the distance between the first paired base, say i, and the next paired base, say j, is 1, and if the distance between the base paired with i, say k, and the base paired with j, say l, is greater than 1, then we have a bulge.

j - i = 1 and $k - l > 1$.

3.3.6 Internal Loops

An internal loop is a face bounded by two interior edges with the two interior edges being separated by two or more exterior edges on both sides. They can be thought of as a special case of bulges, and are extracted in a similar way. In this case, the distance between both the paired bases and the bases paired with them is greater than 1. That is,

$j - i > 1$ and $k - l > 1$.

3.3.7 Bifurcation Loops

A certain caveat that was not mentioned while discussing the extraction of bulges and internal loops is that from the ways that they were extracted, there could be a possibility of *another* base pair appearing between either i and j or k and l. What this would result in is a face bounded by three or more interior edges, and as we have already noted in the previous chapter this is referred to as a bifurcation loop or a multi-loop.

The way to escape this particular problem is to check for the presence of another base pair between i and j or k and l, **before** extracting the bulge or internal loop.

The code for extracting the bifurcation loops, bulges, internal loops, and stacks is as follows:

```
# Find a "face" : a region bounded by an interior edge (base pair)
# and another either interior edge or exterior edge (phosphodiester bond)
while base < len(bpseq):
    # Find the first base pair
    while base < len(bpseq) and bpseq[base][2] == 0:
        base += 1
    # Find the second base pair
    end = base + 1
    while end < len(bpseq) and bpseq[end][2] == 0:
        end += 1

    if bpseq[base][0] in checkedbases or bpseq[end][0] in checkedbases:
        base = end
        continue
    checkedbases.add(bpseq[base][2])
    checkedbases.add(bpseq[end][2])

    if end == len(bpseq) or base == len(bpseq):
        break

    # check the db notation and if there exists a bp between bpseq[base][2]
    # and bpseq[end][2] then it is a bifurcation loop
    if "(" in db[bpseq[end][2]-1:bpseq[base][2]-1]:
        bifurcationcount += 1
    else:
        i1i2 = bpseq[end][0] - bpseq[base][0]
        j1j2 = abs(bpseq[end][2] - bpseq[base][2])
        if (i1i2 == 1 and j1j2 == 1) or (i1i2 == 2 and j1j2 == 2):
            stackcount += 1
        elif i1i2 > 1 and j1j2 > 1:
            interiorcount += 1
        else:
            bulgecount += 1
    base = end
```

3.4 Implementation and Training

The following section details the implementation of the SVM model and the training process, but before that it is necessary to understand some preliminary concepts. These definitions and concepts are borrowed from the work of Akiyama et al.[15], since I am going to be using the same model in this project.

3.4.1 Preliminaries

The alphabet of the RNA sequence is $\Sigma = \{A, C, G, U\}$, and the set of all possible RNA sequences is denoted by Σ^*. Let s be an RNA sequence of length n, denoted by $|s| = n$, and the ith base of the sequence is denoted by s_i. Let $\Pi(s)$ denote the set of all possible secondary structures of an RNA sequence s, and this particular structure is defined as a triangular matrix of 1s and 0s, where a 1 at the ith row and jth column denotes that the ith base and the jth base are paired, and a 0 denotes that they are unpaired, keeping in mind that pseudoknots are not allowed in this model, and a base can only be paired with one other base.

The feature representation of the secondary structure π is denoted by $\phi(s, \pi)$, and contains the number of occurrences of each feature in the structure.

The scoring function based on the machine learning approach $f(s, \pi)$ is the function that assigns scores to the secondary structure $\pi \in \Pi(s)$ of the RNA sequence $s \in \Sigma^*$. As noted before, we require the weights for each feature to calculate the score, and thus this is what the scoring function looks like:

$$f(s, \pi) = \sum_{k=1}^{K} \lambda_k \phi_k(s, \pi') \tag{3.1}$$

where λ_k is the weight of the kth feature, and K is the total number of features.

3.4.2 Learning Algorithm

To actually learn the feature weights, the model Akiyama et al.[15] proposed and the model that I used is an SSVM[40], which is a variant of the Support Vector Machine(SVM) model. And so, we need to find λ that minimises the following objective function:

$$\mathcal{L}(\lambda) = \sum_{(s,\pi)\in D} \max_{\pi'\in\Pi(s)} (f(s,\pi') + \Delta(\pi,\pi')) - f(s,pi) + C \cdot ||\lambda||_1 \tag{3.2}$$

where D is the training dataset, $\Delta(\pi,\pi')$ is the loss function that measures the difference between the predicted structure π' and the true structure π, and $||\lambda||_1$ is the L_1 norm of the feature weights, and C is the L_1 regularisation parameter. The loss function $\Delta(\pi,\pi')$ is defined as:

$$\Delta(\pi,\pi') = \delta^F N \times (Falsenegatives) + \delta^F P \times (Falsepositives) \tag{3.3}$$

where $\delta^F N$ and $\delta^F P$ are the tunable weights for false negatives and false positives respectively.

Finally lets look at the algorithm to train the model:

```
$\lambda \leftarrow 0$
**repeat**
    **for all** $(s,\pi) \in D$ **do**
        $\pi' \leftarrow \arg\max_{\pi'}[f(s,\pi') + \Delta(\pi,\pi')]$
        **for all** $\lambda_k \in \lambda$ **do**
            $\lambda_k \leftarrow \lambda_k - \eta(\phi_k(s,\pi') - \phi_k(s,\pi) + C \cdot \mathrm{sgn}(\lambda_k))$
        **end for**
    **end for**
**until** all the parameters converge
```

Akiyama et al.[15] mentioned that we could use a Zuker-style dynamic programming algorithm **modified by loss-augmented inference** to calculate the first term [40]. However they do not delve deeper into how that should be done, so let us see how we can use the Zuker algorithm with loss-augmented inference.

3.4.3 Loss-Augmented Inference

Let $Sub(s,\pi')$ denote the set of all substructures of the RNA sequence s that are compatible with the secondary structure π'. The feature representation is denoted by $\phi(s, \pi')$, which can also be broken down into terms of its substructures:

$$\phi(s, \pi') = \sum_{bp \in Sub(s,\pi')} \phi(bp) \tag{3.4}$$

The bp-pair is the outermost closing base pair of the substructure bp, $s_bp = s_ij$ is the subsequence of the RNA sequence s that is enclosed by the base pairs i and j. Taking a closer look at the loss function $\Delta(\pi, \pi')$ in equation 3.3,

$$\begin{aligned}
\Delta(\pi, \pi') &= \delta^F N \times (Falsenegatives) + \delta^F P \times (Falsepositives) \\
&= \delta^F N \sum_{i<j} I(\pi_{ij} = 1) I(\pi'_{ij} = 0) + \delta^F P \sum i < j I(\pi_{ij} = 0) I(\pi'_{ij} = 1) \\
&= \sum_{i<j} \delta^F N \pi_{ij}(1 - \pi'_{ij}) + \delta^F P(1 - \pi_{ij})\pi'_{ij}
\end{aligned}$$

$I(\pi_{ij} = 1)$ is the indicator function that is 1 if the base pair ij is present in the structure π, and 0 otherwise. $I(\pi_{ij} = 1) = \pi_{ij}$ and $I(\pi_{ij} = 0) = 1 - \pi_{ij}$.

Rewriting the first term of the objective function in equation 3.2:

$$\begin{aligned}
f(s, \pi') + \Delta(\pi, \pi') &= \lambda\phi(s, \pi') + \sum_{i<j} \delta^{FN}\pi_{ij}(1 - \pi'_{ij}) + \delta^{FP}(1 - \pi_{ij})\pi'_{ij} \\
&= \sum_{bp \in Sub(s,\pi')} \lambda\phi(bp) + \sum_{i<j} [-\delta^{FN}\pi_{ij} + \delta^{FP}(1 - \pi_{ij})]\pi'_{ij} + \delta^{FN}\pi_{ij} \\
&= \sum_{bp \in Sub(s,\pi')} \lambda\phi(bp) + \sum_{i<j where \pi'_{ij}=1} [-\delta^{FN}\pi_{ij} + \delta^{FP}(1 - \pi_{ij})] + C \\
&= \sum_{bp \in Sub(s,\pi')} [\lambda\phi(bp) - \delta^{FN}\pi_{bp-pair} + \delta^{FP}(1 - \pi_{bp-pair})] + C \\
&= \sum_{bp \in Sub(s,\pi')} [\lambda\phi(bp) + \tau_{bp}] + C
\end{aligned}$$

Therefore:

$$f(s, \pi') + \Delta(\pi, \pi') = \sum_{bp \in Sub(s,\pi')} [\lambda\phi(bp) + \tau_{bp}] + C \tag{3.5}$$

Where $\tau_{bp} = -\delta^{FN}\pi_{bp-pair} + \delta^{FP}(1 - \pi_{bp-pair})$

$$\tau_{bp} = \begin{cases} -\delta^{FN} & \text{if } \pi_{bp-pair} = 1 \\ \delta^{FP} & \text{if } \pi_{bp-pair} = 0 \end{cases} \tag{3.6}$$

and

$$C = \sum_{i<j} \delta^{FN} \pi_{ij} \tag{3.7}$$

using the above equations 3.5, 3.6, and 3.7, we can now use the dynamic programming algorithm using loss-augmented inference.

3.4.4 Training

I used the dynamic programming implementation provided by LatticeAutomation's seqfold package[1] to get the predictions for the secondary structures, and used the learning algorithm to train the model on the TrainSetA dataset.

A high level overview of the training process is provided below:

```
eta = 0.1
C = 0.001
# loop through the training data from index to the end
for i in range(index+1, len(training_data)):
    seq = training_data.iloc[i]["Sequence"]
    db = training_data.iloc[i]["Structure"]
    features = training_data.iloc[i][2:]

    # Training on the ith data point

    predstruct =  dRNA.fold(seq, scores = {})
    preddb = dRNA.dot_bracket(seq, predstruct)
    pred_features = extract_features(predstruct, preddb)

    lambda_w = {k: v - eta * (predfeatures[i] - features[i] + C * np.sign(v))
```

The training process was done on a machine with an Apple M1 chip with 16GB of RAM, and the training took about 2 days to complete.

[1] https://github.com/Lattice-Automation/seqfold

Chapter 4

Results

4.1 Evaluation Metrics

Standard evaluation metrics like exact accuracy, precision and recall do not work well in the context of RNA secondary structure prediction, since it does not make sense to judge a particular model based on the fact that it is not able to predict the entire structure correctly, because it may be the case that most of the predicted base pairs are correct.

And so, the metrics generally used to evaluate the accuracy of prediction are Sensitivity, Specificity, and the Matthews Correlation Coefficient(MCC), a popularly used metric in computational biology[45]. Before delving deeper into the metrics, it is important to understand the concepts of True Positives(TP), True Negatives(TN), False Positives(FP), and False Negatives(FN).

- **True Positives(TP)**: The number of base pairs that are correctly predicted by the model.
- **True Negatives(TN)**: The number of base pairs that are correctly predicted to be unpaired by the model.
- **False Positives(FP)**: The number of base pairs that are incorrectly predicted to be paired by the model.
- **False Negatives(FN)**: The number of base pairs that are incorrectly predicted to be unpaired by the model.

Metric	**Value**
Sensitivity	0.512
Specificity	0.498
F-value	0.50490

Table 4.1: Evaluation of our model on the TestSetA dataset.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{4.1}$$

Specificity, also known as "Selectivity" or "Positive Predictive Value(PPV)" is the proportion of true negatives that are correctly identified by the model.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4.2}$$

MCC is a useful metric to evaluate the quality algorithms as well, it is defined as:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.3}$$

It ranges from -1, i.e the worst possible prediction(TP=TN=0), to 1, i.e the best possible prediction(FP=FN=0)

Another measure used is the F-value which is a harmonic mean of the Sensitivity and Specificity:

$$\text{F-value} = \frac{2 \times \text{Sensitivity} \times \text{Specificity}}{\text{Sensitivity} + \text{Specificity}} \tag{4.4}$$

4.2 Results

The model was trained on the TrainSetA dataset and tested on the TestSetA dataset, and the results obtained are presented in Table 4.1.

The similar metrics for the model proposed by Akiyama et al. are presented in Table 4.2.

Table 4.2: Accuracy for each scoring model: The thermodynamic model (TM), the machine learning-based model (ML) trained with TrainSetA, and the integrated model (TM+ML). By Akiyama et al.

	TestSetA			TestSetB		
Model	SEN	PPV	F	SEN	PPV	F
TM	0.682	0.659	0.670	0.598	0.485	0.536
ML	0.703	0.764	0.732	0.575	0.550	0.563
TM+ML	0.715	0.761	0.737	0.617	0.565	0.590

4.3 Discussion

As is evident from the results, the model proposed by Akiyama et al. outperforms the model that I trained on the TrainSetA dataset. This is because of a few reasons, the most important of which is the fact that the model proposed by Akiyama et al. was trained on a larger dataset, utilising TrainSetB as well, and was able to learn from structurally dissimilar sequences. I decided to only use the TrainSetA dataset because of the computational resources available to me, the choice of the programming language and the time it would take to train the model on a larger dataset, since the time on TrainSetA itself was quite high(2 days).

Akiyama, Sato, and Sakakibara[46] also improved on their earlier work and released another model, aptly named "mxfold", which made a number of improvements in accuracy.

Since the Akiyama model, a number of other machine learning projects have also been proposed, like the SPOT-RNA model by Singh et al.[37], an ensemble model that uses a combination of different machine learning models to predict RNA secondary structures. ContextFold by Zako et al.[47], a model that pioneered the use of rich parameterisation, using parameters orders of magnitude higher than any other model at the time(70,000); RNAFold by ViennaRNA[23], a popular RNA secondary structure prediction tool in the bioinformatics community; and TORNADO by Rivas et al.[39], a general purpose parser to produce grammars of single-sequence RNA secondary structure.

An evaluation of all the models is given in Table 4.3.

We can see how the accuracy falls off when dealing with sequences that are structurally dissimilar. This underscores the importance of training the model on a diverse dataset, to ensure that the model is able to generalise

Table 4.3: Performance metrics for various RNA secondary structure prediction methods. PPV: Positive Predictive Value, SEN: Sensitivity, F: F1-score. Taken from Sato et al.[46]

	Sequence-wise CV			Family-wise CV		
Method	PPV	SEN	F	PPV	SEN	F
MXfold2	0.520	0.682	0.575	0.585	0.710	0.632
SPOT-RNA	0.652	0.578	0.597	0.599	0.619	0.596
TORNADO	0.554	0.609	0.561	0.636	0.638	0.620
ContextFold	0.583	0.595	0.575	0.595	0.539	0.554
RNAfold	0.446	0.631	0.508	0.552	0.720	0.617
My model	0.498	0.512	0.504	0.221	0.342	0.26849

well.

Chapter 5

Conclusion

5.1 Summary

In this project, I proposed a model to predict RNA secondary structures using a machine learning approach, and trained the model on the TrainSetA dataset proposed by Rivas et al.[39]. The model was trained using the SSVM model [40] proposed by Akiyama et al.[15] for this field, and the results were evaluated using the Sensitivity, Specificity, and F-value metrics. In doing so, we derived the feature representation of the secondary structures, and used the dynamic programming algorithm to predict the structures, modified by loss-augmented inference.

5.2 Future Work

There are certain assumptions and simplifications that were made in this project in order to make the model more tractable, and so there are certain ways that the model can be improved. Firstly, the model can be trained on a larger dataset, like the TrainSetB dataset, to ensure that the model is able to generalise well and learn from all possible RNA families. Structural motifs hitherto unexplored in a large number of models should also be incorporated and considered like the pseudoknots.

We also noted how long the model takes to train on a relatively small dataset. This is because of the time taken by the dynamic programming algorithm to predict the secondary structures, based on the weights, running in $O(N^3)$ time, but it was also not helped by the choice of the programming language, Python, which makes it very easy to implement algorithms but is not fast in terms of computation speed. It might be interesting to explore “Mojo” [1], a programming language that aims to combine the ease of use of Python with the speed of performant languages like C/C++.

As mentioned before, more complex models have also arisen in the last few years that have improved on the accuracy of the predictions, and so it would be interesting to explore those approaches along with the probabilistic models that have been proposed in the past.

[1] https://www.modular.com/max/mojo

Bibliography

[1] I. Tinoco and C. Bustamante, "How RNA folds," *Journal of Molecular Biology*, vol. 293, no. 2, pp. 271–281, 1999, ISSN: 0022-2836. DOI: https://doi.org/10.1006/jmbi.1999.3001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022283699930012.

[2] J. M. Berg, J. L. Tymoczko, and L. Stryer, *Biochemistry*, 5th. WH Freeman and Company, 2002, pp. 118–119, 781–808, ISBN: 978-0-7167-4684-3.

[3] H. F. Noller, V. Hoffarth, and L. Zimniak, "Unusual resistance of peptidyl transferase to protein extraction procedures," *Science*, vol. 256, no. 5062, pp. 1416–1419, 1992. DOI: 10.1126/science.1604315. eprint: https://www.science.org/doi/pdf/10.1126/science.1604315. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1604315.

[4] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, "Molecular biology of the cell," in 4th, From RNA to Protein, New York: Garland Science, 2002, ch. tRNA Molecules Match Amino Acids to Codons in mRNA. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK26829/.

[5] M. Shigematsu and Y. Kirino, "Trna-derived short non-coding rna as interacting partners of argonaute proteins," *Gene Regul Syst Bio*, vol. 9, pp. 27–33, Sep. 2015. DOI: 10.4137/GRSB.S29411.

[6] W.-T. Wang, C. Han, Y.-M. Sun, and et al., "Noncoding RNAs in cancer therapy resistance and targeted drug development," *Journal of Hematology & Oncology*, vol. 12, p. 55, 2019. DOI: 10.1186/s13045-019-0748-z. [Online]. Available: https://doi.org/10.1186/s13045-019-0748-z.

[7] B. Onoa and I. Tinoco Jr, “RNA folding and unfolding,” *Current Opinion in Structural Biology*, vol. 14, no. 3, pp. 374–379, Jun. 2004. DOI: 10.1016/j.sbi.2004.04.001.

[8] P. Johnsson, L. Lipovich, D. Grandér, and K. V. Morris, “Evolutionary conservation of long non-coding rnas; sequence, structure, function,” *Biochimica et Biophysica Acta (BBA)*, vol. 1840, no. 3, pp. 1063–1071, Mar. 2014. DOI: 10.1016/j.bbagen.2013.10.035.

[9] J. E. McCarthy and C. Gualerzi, “Translational control of prokaryotic gene expression,” *Trends in Genetics*, vol. 6, no. 3, pp. 78–85, Mar. 1990. DOI: 10.1016/0168-9525(90)90098-Q.

[10] A. C. Palmenberg and J.-Y. Sgro, “Topological organization of picornaviral genomes: Statistical prediction of RNA structural signals,” *Seminars in Virology*, vol. 8, no. 3, pp. 231–241, 1997, ISSN: 1044-5773. DOI: 10.1006/smvy.1997.0126. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1044577397901268.

[11] J. R. Fresco, B. M. Alberts, and P. Doty, “Some molecular details of the secondary structure of ribonucleic acid,” *Nature*, vol. 188, pp. 98–101, 1960. DOI: 10.1038/188098a0. [Online]. Available: https://doi.org/10.1038/188098a0.

[12] Q. Zhao, Z. Zhao, X. Fan, Z. Yuan, Q. Mao, and Y. Yao, “Review of machine learning methods for RNA secondary structure prediction,” *PLoS Computational Biology*, vol. 17, no. 8, e1009291, 2021. DOI: 10.1371/journal.pcbi.1009291. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1009291.

[13] M. Zuker and P. Stiegler, “Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information,” *Nucleic Acids Research*, vol. 9, no. 1, pp. 133–148, Jan. 1981. DOI: 10.1093/nar/9.1.133.

[14] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995. DOI: 10.1007/BF00994018. [Online]. Available: https://doi.org/10.1007/BF00994018.

[15] M. Akiyama, K. Sato, and Y. Sakakibara, “A max-margin training of RNA secondary structure prediction integrated with the thermodynamic model,” *Journal of Bioinformatics and Computational Biology*, vol. 16, no. 06, 2018.

[16] R. W. Holley, J. Apgar, G. A. Everett, J. T. Madison, M. Marquisee, S. H. Merrill, J. R. Penswick, and A. Zamir, “Structure of a ribonucleic acid,” *Science*, vol. 147, no. 3664, pp. 1462–1465, 1965. DOI: 10.1126/science.147.3664.1462. eprint: https://www.science.org/doi/pdf/10.1126/science.147.3664.1462. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.147.3664.1462.

[17] F. H. Crick, “Codon–anticodon pairing: The wobble hypothesis,” *Journal of Molecular Biology*, vol. 19, no. 2, pp. 548–555, Aug. 1966. DOI: 10.1016/s0022-2836(66)80022-0.

[18] T. Kamada and S. Kawai, “An algorithm for drawing general undirected graphs,” *Information Processing Letters*, vol. 31, no. 1, pp. 7–15, 1989, ISSN: 0020-0190. DOI: https://doi.org/10.1016/0020-0190(89)90102-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0020019089901026.

[19] P. Danaee, M. Rouches, M. Wiley, D. Deng, L. Huang, and D. Hendrix, “Bprna: Large-scale automated annotation and analysis of RNA secondary structure,” *Nucleic Acids Research*, vol. 46, no. 11, pp. 5381–5394, Jun. 2018. DOI: 10.1093/nar/gky285.

[20] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using NetworkX,” in *Proceedings of the 7th Python in Science Conference (SciPy 2008)*, SciPy, 2008. [Online]. Available: https://aric.hagberg.org/papers/hagberg-2008-exploring.pdf.

[21] I. Tinoco, O. C. Uhlenbeck, and M. D. Levine, “Estimation of secondary structure in ribonucleic acids,” *Nature*, vol. 230, pp. 362–367, 1971. DOI: 10.1038/230362a0. [Online]. Available: https://doi.org/10.1038/230362a0.

[22] S. M. Freier, R. Kierzek, J. A. Jaeger, N. Sugimoto, M. H. Caruthers, T. Neilson, and D. H. Turner, “Improved free-energy parameters for predictions of RNA duplex stability,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 83, no. 24, pp. 9373–9377, Dec. 1986. DOI: 10.1073/pnas.83.24.9373.

[23] R. Lorenz, S. H. Bernhart, C. Höner zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker, “ViennaRNA package 2.0,” *Algorithms for Molecular Biology*, vol. 6, no. 1, p. 26, 2011. DOI: 10.1186/1748-7188-6-26.

[24] E. Rivas and S. R. Eddy, “A dynamic programming algorithm for RNA structure prediction including pseudoknots,” *Journal of Molecular Biology*, vol. 285, no. 5, pp. 2053–2068, Feb. 1999. DOI: 10.1006/jmbi.1998.2436.

[25] D. W. Staple and S. E. Butcher, “Pseudoknots: RNA structures with diverse functions,” *PLoS Biology*, vol. 3, no. 6, e213, Jun. 2005. DOI: 10.1371/journal.pbio.0030213.

[26] K. Rietveld, R. Van Poelgeest, C. W. Pleij, J. H. Van Boom, and L. Bosch, “The trna-like structure at the 3’ terminus of turnip yellow mosaic virus RNA. differences and similarities with canonical tRNA,” *Nucleic Acids Research*, vol. 10, no. 6, pp. 1929–1946, Mar. 1982. DOI: 10.1093/nar/10.6.1929.

[27] E. Westhof, “Twenty years of RNA crystallography,” *RNA*, vol. 21, no. 4, pp. 486–487, Apr. 2015. DOI: 10.1261/rna.049726.115.

[28] B. Fürtig, C. Richter, J. Wöhnert, and H. Schwalbe, “NMR Spectroscopy of RNA,” *ChemBioChem*, vol. 4, no. 10, pp. 936–962, 2003.

[29] R. Nussinov, G. Pieczenik, J. R. Griggs, and D. J. Kleitman, “Algorithms for loop matchings,” *SIAM Journal on Applied Mathematics*, vol. 35, no. 1, pp. 68–82, Jul. 1978. [Online]. Available: https://www.jstor.org/stable/2101031.

[30] J. Kim, A. E. Walter, and D. H. Turner, “Thermodynamics of coaxially stacked helixes with ga and cc mismatches,” *Biochemistry*, vol. 35, no. 43, pp. 13 753–13 761, Oct. 1996. DOI: 10.1021/bi960913z. [Online]. Available: https://doi.org/10.1021/bi960913z.

[31] S. J. Schroeder and D. H. Turner, “Optical melting measurements of nucleic acid thermodynamics,” *Methods in Enzymology*, vol. 468, pp. 371–387, 2009. DOI: 10.1016/S0076-6879(09)68017-4.

[32] J. M. Pipas and J. E. McMahon, “Method for predicting RNA secondary structure,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 72, no. 6, pp. 2017–2021, Jun. 1975. DOI: 10.1073/pnas.72.6.2017.

[33] M. Zuker, “Mfold web server for nucleic acid folding and hybridization prediction,” *Nucleic Acids Res*, vol. 31, no. 13, pp. 3406–3415, Jul. 2003. DOI: 10.1093/nar/gkg595.

[34] N. R. Markham and M. Zuker, "Unafold: Software for nucleic acid folding and hybridization," in *Methods in Molecular Biology™*, ser. MIMB. Springer, 2008, vol. 453.

[35] D. H. Turner and D. H. Mathews, "NNDB: The nearest neighbor parameter database for predicting stability of nucleic acid secondary structure," *Nucleic Acids Research*, vol. 38, no. suppl_1, pp. D280–D282, Jan. 2010. DOI: 10.1093/nar/gkp892. [Online]. Available: https://doi.org/10.1093/nar/gkp892.

[36] C. B. Do, D. A. Woods, and S. Batzoglou, "Contrafold: RNA secondary structure prediction without physics-based models," *Bioinformatics*, vol. 22, no. 14, e90–e98, 2006. DOI: 10.1093/bioinformatics/btl246.

[37] 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," *Nature Communications*, vol. 10, pp. 1–10, 2019.

[38] R. Durbin, S. R. Eddy, A. Krogh, and G. J. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge, UK: Cambridge University Press, 1998.

[39] E. Rivas, R. Lang, and S. R. Eddy, "A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more," *RNA*, vol. 18, no. 2, pp. 193–212, Feb. 2012. DOI: 10.1261/rna.030049.111.

[40] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, Sep. 2005.

[41] E. Rivas, "The four ingredients of single-sequence RNA secondary structure prediction. A unifying perspective," *RNA Biology*, vol. 10, no. 7, pp. 1185–1196, Jul. 2013. DOI: 10.4161/rna.24971.

[42] M. Andronescu, A. Condon, H. H. Hoos, D. H. Mathews, and K. P. Murphy, "Efficient parameter estimation for RNA secondary structure prediction," *Bioinformatics*, vol. 23, no. 13, pp. i19–i28, Jul. 2007. DOI: 10.1093/bioinformatics/btm223. [Online]. Available: https://doi.org/10.1093/bioinformatics/btm223.

[43] ——, "Computational approaches for RNA energy parameter estimation," *RNA*, vol. 16, no. 12, pp. 2304–2318, Dec. 2010. DOI: 10.1261/rna.1950510.

[44] Z. J. Lu, J. W. Gloor, and D. H. Mathews, "Improved RNA secondary structure prediction by maximizing expected pair accuracy," *RNA*, vol. 15, no. 10, pp. 1805–1813, Oct. 2009. DOI: 10.1261/rna.1643609.

[45] B. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, pp. 442–451, Oct. 1975.

[46] K. Sato, M. Akiyama, and Y. Sakakibara, "Rna secondary structure prediction using deep learning with thermodynamic integration," *Nature Communications*, vol. 12, p. 941, 2021, Published: 11 February 2021. DOI: 10.1038/s41467-021-21281-9. [Online]. Available: https://doi.org/10.1038/s41467-021-21281-9.

[47] S. Zakov, Y. Goldberg, M. Elhadad, and M. Ziv-ukelson, "Rich parameterization improves rna structure prediction," *Journal of Computational Biology*, vol. 18, no. 11, 2011, Published Online: 10 November 2011. DOI: 10.1089/cmb.2011.0184. [Online]. Available: https://doi.org/10.1089/cmb.2011.0184.

Appendix A

Appendix

A.1 Secondary Structures

A.1.1 Bacteria, *Anabaena variabilis*

Sequence: AAGCCUGGGCCCGUGCGGCUCGGACGCCCGAACCGUGUCAGGACCUGACGG UAGCAGCACUAAGGGAUGCUCUGGGCAGGCGCGUGGUUCCGGGUUUUUUC
DB Notation: ((.((..(.(.(((((.(((((((((((((.....((((....(((....)))....))))...))).)).)))))))..))))).)..)..)).)).....

A.1.2 Archaea, *Haloquadratum walsbyi*

Sequence: GAUUCCGUAAGUUCGGAUUUGAGGCGGCCAGAGCGGCAGGGAAACACCUGUACCCAU CCCGAACACAGUGGUUAAGUCUGCAAGCGUUGGAGCAC GUACUGGAGUGAGAGAUCCUCUGGGAGCGCUUCAUCGC CGCCUU
DB Notation: ......................((((((....((((((((.... (((((((.............))))..)))...)))))).)).(((((((...((((((((.((....)))))))))))..))))))).))))))..

A.1.3 Bacteria, *Microbacterium*

Sequence: UGAGUAAUGUCUGGGAAACUGCCUGAUGGAGGGGGAUAACUACUGGAAAC GGUAGCUAAUACCGCAUAACGUCGCAAGACCAAAGAGGGGGACCUUCGGGCCUCUUGCCAUCAGAU GUGCCCAGAUGGGAUUAGCUAGUAGGUGGGGUAACGGCUCACCUAGGCGACGAUCCCUAGCUGGUC UGAGAGGAUGACCAGCCACACUGGAACUGAGACACGGUCCAGACUCCUACGGGAGGCAGCAGUGGG GAAUAUUGCACAAUGGGCGCAAGCCUGAUGCAGCCAUGCCGCGUGUAUGAAGAAGGCCUUCGGGUU GUAAAGUACUUUCAGCGGGGAGGAAGGUGCUGAGGUUAAUAACCUCAGCAAUUGACGUUACCCGCA GAAGAAGCACCGGCUAACUCCGUGCCAGCAGCCGCGGUAAUACGGAGGGUGCAAGCGUUAAUCGGA AUUACUGGGCGUAAAGCG
DB Notation: ((......((((((((.((...(((((((.((((....(((((((....))))))) .....))))......(((....)))....((..................)).))))))))..))))))))))(((..
(.(((..((((((((.......)))))))))))).....))))..((((((((....))))...)))))))..((((((..........)))))).((((....))))................(.(((...((((((....)))))).)))).............
((((......((((....))))......))))).[.((((((((...(.......(((((.......)))))........)....)))))))..])..((((([[[...((((((.....((.]]]))......))))))))))..............................