RNA Sequence Classification using Secondary Structure Fingerprints, Sequence-Based Features, and Deep Learning

by

Kevin Sutanto

A thesis
submitted to the University of Ottawa
in partial fulfillment of the
thesis requirement for the degree of
Master of Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

RNAs are involved in different facets of biological processes; including but not limited to controlling and inhibiting gene expressions, enabling transcription and translation from DNA to proteins, in processes involving diseases such as cancer, and virus-host interactions. As such, there are useful applications that may arise from studies and analyses involving RNAs, such as detecting cancer by measuring the abundance of specific RNAs, detecting and identifying infections involving RNA viruses, identifying the origins of and relationships between RNA viruses, and identifying potential targets when designing novel drugs.

Extracting sequences from RNA samples is usually not a major limitation anymore thanks to sequencing technologies such as RNA-Seq. However, accurately identifying and analyzing the extracted sequences is often still the bottleneck when it comes to developing RNA-based applications.

Like proteins, functional RNAs are able to fold into complex structures in order to perform specific functions throughout their lifecycle. This suggests that structural information can be used to identify or classify RNA sequences, in addition to the sequence information of the RNA itself. Furthermore, a strand of RNA may have more than one possible structural conformations it can fold into, and it is also possible for a strand to form different structures *in vivo* and *in vitro*. However, past studies that utilized secondary structure information for RNA identification purposes have relied on one predicted secondary structure for each RNA sequence, despite the possible one-to-many relationship between a strand of RNA and the possible secondary structures. Therefore, we hypothesized that using a representation that includes the multiple possible secondary structures of an RNA for classification purposes may improve the classification performance.

We proposed and built a pipeline that produces secondary structure fingerprints given a sequence of RNA, that takes into account the aforementioned multiple possible secondary structures for a single RNA. Using this pipeline, we explored and developed different types of secondary structure fingerprints in our studies. A type of fingerprints serves as high-level topological representations of the RNA structure, while another type represents matches with common known RNA secondary structure motifs we have curated from databases and the literature. Next, to test our hypothesis, the different fingerprints are then used with deep learning and with different datasets, alone and together with various sequence-based features, to investigate how the secondary structure fingerprints affect the classification performance.

Finally, by analyzing our findings, we also propose approaches that can be adopted by future studies to further improve our secondary structure fingerprints and classification performance.

Acknowledgements

First and foremost, I would like to thank my thesis supervisor, Dr. Marcel Turcotte, for his guidance, feedback, and supervision throughout my work. In addition to his supervision, his courses had also introduced me to various concepts of biology I was not taught about throughout my previous educational experience, which are fundamental to this work. His passion for biology and bioinformatics has been inspirational and motivating to me.

I would also like to thank Compute Canada for the computing facilities they have provided to perform my research, and the Natural Sciences and Engineering Research Council for the funding they have provided to our research group.

Without all of these, this thesis would not have been possible.

Table of Contents

List of Tables viii

List of Figures ix

Glossary xii

1 Introduction 1

1.1 Motivation 1

1.2 Biological Question 3

1.3 Problem Statements 11

1.4 Contributions 11

2 Literature Review 13

2.1 Sequence Information for RNA Identification and Classification 13

2.1.1 K-mer Overview 13

2.1.2 K-mer in Prior Studies 14

2.1.3 Non-continuous K-mer Overview 16

2.1.4 Non-continuous K-mer in Prior Studies 17

2.2 Secondary Structure Information for RNA Identification and Classification 20

2.2.1 Secondary Structure Information in Prior Studies 20

2.2.2 Possible Limitation of Prior Studies 21

2.3 Graph Representation of Secondary Structure . . . 22
2.3.1 "RNA-As-Graphs" . . . 23
2.4 Deep Learning . . . 27
2.4.1 Supervised Learning and Artificial Neural Network . . . 27
2.4.2 Deep Neural Network . . . 28
2.4.3 Hyperparameters . . . 29
2.4.4 Classification Performance Measurement . . . 30
2.4.5 Deep Learning in Prior Studies . . . 33

3 Methods **34**
3.1 Graph-Based Secondary Structure Fingerprints . . . 34
3.2 Common Secondary Structure Motif Fingerprints . . . 42
3.3 K-mers . . . 51
3.4 Skip-mers . . . 52
3.5 Deep Learning . . . 53

4 Datasets **59**
4.1 Non-Coding RNAs from Rfam 14.1 . . . 59
4.1.1 Overview . . . 59
4.1.2 Deep Learning Features . . . 60
4.1.3 Deep Learning Configuration . . . 61
4.1.4 Training Hyperparameters and Evaluation . . . 61
4.2 RNA Virus Sequences and Their Host Species from NCBI Virus . . . 62
4.2.1 Overview . . . 62
4.2.2 Additional CD-hit Reduced Dataset . . . 63
4.2.3 Deep Learning Features . . . 64
4.2.4 Deep Learning Configuration . . . 66
4.2.5 Training Hyperparameters and Evaluation . . . 66

5 Results and Discussion **68**

5.1 Non-Coding RNAs from Rfam 14.1 68

5.1.1 Results Overview 68

5.1.2 Comparison of the Different Secondary Structure Fingerprint Types 71

5.1.3 Combining Different Secondary Structure Features 71

5.1.4 Sequence-Based Features and Secondary Structure Based Features . 76

5.2 RNA Virus Sequences and their Host Species from NCBI Virus 77

5.2.1 Primary Results Overview 77

5.2.2 Reduced Dataset Results Overview 78

5.2.3 Comparison of the Different Secondary Structure Fingerprint Types 84

5.2.4 Combining Free Energy Values to Produce the Secondary Structure Fingerprints 85

5.2.5 Combining Different Feature Types 86

5.2.6 Best Performing Feature Set is Different for the Reduced Datasets . 87

6 Limitation **88**

6.1 Limited Sequence Length 88

6.2 No Distinction between Local and Global Matches by the Secondary Structure Fingerprints 89

6.3 Use of RNAMotif Default Minimum Length for the RAG-based Secondary Structure Fingerprints 89

6.4 Complete Viral Sequence Requirement 89

7 Future Work **91**

7.1 Improved Secondary Structure Fingerprints 91

7.1.1 Produce and Include Different Score Variations to Build Better Performing Fingerprints 91

7.1.2 Include Information on Local vs. Global Matches 92

7.1.3 Include Positional Information of Secondary Structure Matches . . 92

7.2 More Efficient Secondary Structure Fingerprints Pipeline 93
7.2.1 Use Smaller but More Descriptors 93
7.2.2 RNAMotif Substitution 93
7.3 Classification with Partial Sequences 94

8 Conclusion **95**

References **98**

List of Tables

2.1 Different metrics to measure classification performance. 31

4.1 Feature sets used for the virus-host classification problem 65

5.1 10-fold validation accuracy and standard errors of the RNA classification problem using Rfam 14.1 dataset 69

5.2 10-fold validation precision and standard errors of the RNA classification problem using Rfam 14.1 dataset 69

5.3 10-fold validation recall and standard errors of the RNA classification problem using Rfam 14.1 dataset 70

5.4 10-fold validation accuracy and standard errors of the RNA virus host classification problem 79

5.5 10-fold validation precision and standard errors of the RNA virus host classification problem 80

5.6 10-fold validation recall and standard errors of the RNA virus host classification problem 81

5.7 Test set accuracy of each feature set for the RNA virus host classification problem, using a reduced dataset that excludes sequences with $> 90\%$ similarity 82

5.8 Test set accuracy of each feature set for the RNA virus host classification problem, using a reduced dataset that excludes sequences with $> 80\%$ similarity 83

List of Figures

1.1 Basic structural elements of an RNA 6
1.2 Pseudoknot-free base pair interactions 7
1.3 Pseudoknotted base pair interactions 7
1.4 Secondary structure of a ribosomal RNA consisting of the basic structural elements 8
1.5 Secondary structure of a Zika virus single-stranded RNA involving pseudoknots 9

2.1 An example configuration of non-continuous k-mer with alternating matching and skipped characters 18
2.2 Different secondary structure elements and their tree graphs 25
2.3 Different secondary structure elements and their dual graphs 26

3.1 RNAMotif process: From a secondary structure descriptor and a sequence, to secondary structure matches 36
3.2 An example of a RNAMotif descriptor describing a hairpin loop 37
3.3 Example matches produced by RNAMotif using the descriptor in Figure 3.2 given an input sequence 37
3.4 A sample traversal of a RAG representation and its secondary structure . . 38
3.5 Another sample traversal of the same RAG representation and its secondary structure 39
3.6 From RAG representations to secondary structure fingerprints 41
3.7 Produced heatmap from parsing the bpRNA database illustrating the distribution of different bulge lengths on each side 43

3.8 Produced heatmap from parsing the bpRNA database illustrating the distribution of different internal loop lengths on each side 43
3.9 Produced heatmap from parsing the bpRNA database illustrating the distribution of different internal loop length combinations considering both sides at the same time 44
3.10 Produced heatmap from parsing the bpRNA database illustrating the distribution of continuous paired nucleotides lengths of stems on each side . . 45
3.11 Produced heatmap from parsing the bpRNA database illustrating the distribution of different lengths of different types of loops 45
3.12 Produced heatmap from parsing the bpRNA database illustrating the frequency distribution of different secondary structure elements with their corresponding lengths on each side 46
3.13 Produced heatmap from parsing the bpRNA database illustrating the frequency distribution of all secondary structure elements and their corresponding lengths, relative to each other 47
3.14 Accuracy and loss plots of the RAG-based fingerprints throughout model training in the non-coding RNA study 54
3.15 Accuracy and loss plots of the short motifs based fingerprints without wobble throughout model training in the non-coding RNA study 54
3.16 Accuracy and loss plots of the short motifs based fingerprints with wobble throughout model training in the non-coding RNA study 55
3.17 Accuracy and loss plots using all secondary structure fingerprints combined throughout model training in the non-coding RNA study 55
3.18 Accuracy and loss plots using all secondary structure fingerprints and 4-mer combined throughout model training in the non-coding RNA study 56
3.19 Accuracy and loss plots using 4-mer combined throughout model training in the RNA virus study 56
3.20 Accuracy and loss plots using match-1-skip-1 skip-mer of length 9 throughout model training in the RNA virus study 57
3.21 Accuracy and loss plots using common motifs based secondary structure fingerprints throughout model training in the RNA virus study 57

4.1 Deep learning architecture for the Rfam 14.1 dataset 62

5.1 Resulting confusion matrices of fold 1 to 4 when all of the secondary structure fingerprints are used together as features for the RNA classification problem of the Rfam 14.1 dataset 73

5.2 Resulting confusion matrices of fold 5 to 8 when all of the secondary structure fingerprints are used together as features for the RNA classification problem of the Rfam 14.1 dataset 74

5.3 Resulting confusion matrices of fold 9 to 10 when all of the secondary structure fingerprints are used together as features for the RNA classification problem of the Rfam 14.1 dataset 75

Glossary

AUC area under the curve, a metric used with the receiver operating characteristics curve to assess classification performance

BLSTM bidirectional long short-term memory, a type of LSTM that takes into account relationships/dependencies between different positions of sequential data in both directions

cis-reg cis-regulatory, a type of ncRNA whose role is to control the transcription process

CNN convolutional neural network, an architecture of a neural network typically used to detect useful patterns in 2 dimensional data such as images

descriptor one of the inputs to the RNAMotif program, describes a secondary structure to search for

DNA deoxyribonucleic acid, one of the macromolecules in organisms that carries genetic information

feature set a set containing one or more types of deep learning input features covered in our studies

lncRNA long non-coding RNA, a type of ncRNA

LOOCV leave-one-out cross-validation; a method to evaluate classification performance of the model, where one sample is excluded from training and used to test the model instead in a single iteration, and the process repeats until all of the samples have been used as the test sample in the different iterations

LSTM long short-term memory, an architecture of a neural network that takes into account relationships/dependencies between different positions of sequential data

miRNA micro RNA; a type of ncRNA, they are short in length

ncRNA non-coding RNA, a type of RNA whose transcript does not code for proteins

ORF open reading frame, parts of DNA or RNA that indicate that they may be used to encode proteins

RNA ribonucleic acid, one of the macromolecules in organisms

rRNA ribosomal RNA, a type of ncRNA

snoRNA small nucleolar RNA, a type of ncRNA

snRNA small nuclear RNA, a type of ncRNA

sRNA small RNA, a type of ncRNA

ssRNA single-stranded RNA

tRNA transfer RNA, a type of ncRNA which takes part in the translation process by carrying the amino acids needed to build proteins

Chapter 1

Introduction

A higher proportion of the human DNA codes for non-coding RNA instead of proteins [117]. These RNAs play a wide variety of roles in biological processes; from being involved in the protein-coding process itself by translating coding/messenger RNA into amino acids [2], to regulating and controlling various "cellular processes" including in diseases such as cancer [6, 26, 27], and even forming "regulatory networks" in the nervous system [117]. RNA also serves as the building blocks of viruses in the case of RNA viruses, such as the widespread HIV [21], the SARS-CoV [63], and the recent SARS-CoV-2 of the COVID-19 pandemic [118]. All of these are possible for RNA molecules, thanks to their capability to not only carry genetic information, but also to form structures on their own and interact with other molecules directly [151].

As RNA is involved in many facets of biological lives, studies on RNA can result in many practical applications, including drug targeting [150], gene expression inhibition [85], cancer detection [22, 92], cancer prognosis [22], and detection of viral infection and virus identification [22].

1.1 Motivation

Many of the RNA-related applications, including the previously mentioned cancer and viral detection [22], stem from reading the RNA sequences, typically using techniques like RNA-seq [149]. Although reading the raw sequences itself is no longer a bottleneck, there is still room for improvements when it comes to analysis, processing, and identification of the raw sequences. This is especially true in cases where the reference/annotated genome

is not yet available to compare and map the raw sequences to (e.g. reading novel or unknown sequences), and thus computational techniques are required to identify the read sequences [34].

This room for contribution in identifying RNA sequences on their own without reference genomes is the primary motivation and focus of this thesis. In particular, this thesis focuses on building and exploring different types of sequence representations, which are then used with deep learning techniques to identify or classify RNA sequences.

There are two main reasons that motivated the use of sequence representations or fingerprints, instead of the sequence themselves. First, there are cases where RNA structures are conserved despite changes in the sequences, as long as the formed structures remain functional [143, 151]. Therefore, some of the representations should capture the conserved structural information, which may imply conserved function and the identity of the RNA, even if the underlying sequence changes. Second, using the entire sequences with machine and deep learning will require significantly larger amount of memory and computational resources for longer sequences, which would in turn significantly limit the maximum length of sequence that can be included in our studies with a reasonable amount of computational resources. On the other hand, using representations of sequences with deep learning require significantly less memory, and thus longer sequences can be included in the studies. As the results of our studies depend on these sequence representations, performance of the different types of representations is also evaluated in this thesis.

When it comes to using RNA secondary structure (see Sections 1.2 and 2.2) information for RNA classification and identification purposes, the previously proposed approaches have mostly relied on secondary structure prediction tools [28, 46, 113]. For each RNA sequence, the prediction tool is used to obtain a single most likely secondary structure, which is then used as features for classification/identification [28, 46, 113]. However, since it is possible for a strand of RNA to have more than one secondary structure conformation [140], and as a strand may form different structures *in vivo*, and *in vitro* or *in silico* [59, 134, 140]; we decided to devise our approach such that information on these additional possible secondary structure conformations for a strand of RNA would be taken into account (in contrast to the approaches proposed by prior related studies), and therefore can be utilized as features for RNA classification/identification purposes.

Meanwhile, our motivation to use deep learning in our studies is due to its demonstrated ability to automatically extract abstract features [108], and reliably produce inference even when the relationship between the input features and correct output inference is unclear [7, 134].

1.2 Biological Question

RNA is one of the macromolecules that make up life [78]. One of its functions is to carry genetic information, which is also a function of the DNA molecules. Similar to DNA molecules, the information is carried and conveyed by the contiguous nucleotides that make up a strand/sequence of RNA; and each nucleotide at each position within the sequence can be one of the four possible bases: adenine, cytosine, guanine, and uracil [74]. Therefore, a strand of RNA can be represented as a string, which primary possible characters at each position are "A", "C", "G", and "U" (representing adenine, cytosine, guanine, and uracil respectively at their corresponding position in the RNA strand).

As sequences are represented by strings, bioinformatics studies involving RNA sequences typically involve analyses of these strings. This means that string representation techniques, such as the "n-grams" [82], are applicable and have been used for sequence analyses.

Among others, two examples of biologically relevant information that can be inferred from the string representations are: 1) evolutionary relationship and distance from string similarity, and 2) identification of sequence motifs from string patterns. Natural biological processes often involve sequence pattern identification as well; for example, immune responses may be triggered by the recognition of "pathogen nucleic acids" [47], and certain microRNA molecules (a type of non-coding RNA) involved in "gene regulation" were found to have specific sequence patterns in order to recognize and bind to their complementary targets [86]. Therefore, checking for the existence and/or prevalence of string patterns may aid in identification and classification of RNA sequences.

In addition to carrying genetic information itself, unlike DNA but like protein, RNA molecules are also capable of catalysis [24] – that is, taking part in the chemical reactions within living organisms. This is possible as RNA molecules, again similar to peptides or proteins, are able to fold into structures, starting from the simpler secondary structures that can be represented in 2D, to a more complex 3D structure [140].

The nucleotides that make up a strand of RNA contribute to the formation of secondary structures. Each nucleotide in the strand can either pair up and form stable hydrogen bonds with another nucleotide in the strand [53], or they can be unpaired. Each paired nucleotide participates in only one pair. The structure of the RNA strand can then be formed due to the fact that the paired nucleotides interact with another nucleotide from a different position in the strand, whereas the unpaired nucleotides do not interact with any other nucleotide.

Typically, the interactions are formed by canonical base pairs, also known as Watson-

Crick base pairs [88, 106] – these pairs are adenine (A) and uracil (U), and guanine (G) and cytosine (C) [88]. In addition to these canonical pairs, non-canonical pairs are also possible [88, 106]. One of the most common non-canonical pairs is guanine (G) and uracil (U) [88], also known by the name of "G·U wobble" pair [146]. As the name suggests, the wobble pair is not as stable/strong as the former pairs [146]. However, due to their higher prevalence compared to other non-canonical pairs [88], the pair is allowed to interact in one of our studies involving the production of secondary structure fingerprints from a sequence (see Section 3.2).

Given these possible pairs to form a secondary structure, it is clear that for a strand of RNA, there may be more than one possible secondary structure that can be formed by the nucleotides. However, the stability of each of the possible structures may differ from one another. Stability is often measured using free energy ($\frac{kcal}{mol}$) [140]. The nearest-neighbor model is the most commonly used model to compute the free energy, see [39] for the limitations of this model. A possible secondary structure with lower free energy is more stable than an alternative possible structure with a higher free energy. Therefore, among several possibilities of secondary structures for a strand of RNA, the structure with the least free energy is often the correct structure.

However, *in vivo*, other factors such as the "RNA chaperones" may influence the RNA folding process so that the structure with the least free energy is not always the preferred form [59,134]. Thus, studies that involve identification of RNA structures and/or structure-related functions should also take possible secondary structures into account, in addition to the secondary structures with the lowest free energy *in vitro* or *in silico*.

In addition to free energy, other metrics have also been used to measure the likeliness of a secondary structure given a sequence of RNA. For example, a metric measures the weighted average distance between structures from different parts of the RNA; and the lower the average distance, the better (also referred to as "ensemble centroids") [43].

Although the secondary structure of a strand of RNA may be complex, especially for strands with longer sequences, the secondary structure can be broken down into the RNA "basic secondary structure elements", which are: helix, loop, bulge, and junction [140]. A helix is made of consecutive nucleotides on one side paired with their corresponding nucleotides on the other side, forming two sides of nucleotides interacting with one another. A loop is formed when there are unpaired nucleotides on both and opposite sides of a helix (internal loop), or at the end of a helix (hairpin loop). Meanwhile, a bulge is formed when there are unpaired nucleotides only on one of the two sides of a helix; creating a "bulge" on that side at the part with the unpaired nucleotides, as that aforementioned part is not sticking to the other side. Finally, a junction is formed when 2 or more separate helices

join – in other words, a junction is the centre where multiple, separate helices connect. Bulges and internal loops can be considered as 2-way junctions [14]. These basic structural elements are illustrated in Figure 1.1. An example of an actual complex RNA structure made up of these elements is shown in Figure 1.4. As RNA secondary structures comprise of the basic structural elements, computational representations of secondary structures can be made up of these elements.

Usually, the interactions between the nucleotides at different positions along the sequence do not "cross" another interaction (e.g. if the sequence of nucleotides is represented as a straight line, a nucleotide at an inner position at one side of the line interacts with another inner nucleotide at the other side, while an outer nucleotide at one side interacts with an outer nucleotide at the other side; as illustrated in Figure 1.2) [114,155]. However, there are cases where the interactions "cross" another interaction (e.g. outer nucleotides at the left side of the sequence interact with inner nucleotides at the right side, while the outer nucleotides at the right side interact with the inner nucleotides at the left side, as depicted by Figure 1.3), forming an RNA structural element known as "pseudoknot" [114, 155]. An example of RNA secondary structure involving pseudoknots in addition to the basic structural elements is shown in Figure 1.5. As pseudoknots may have functional roles in RNAs [17, 52, 84, 132], information on pseudoknots may be useful for identification and classification of functional RNAs. However, secondary structure prediction and matching involving pseudoknots requires more computational resources.

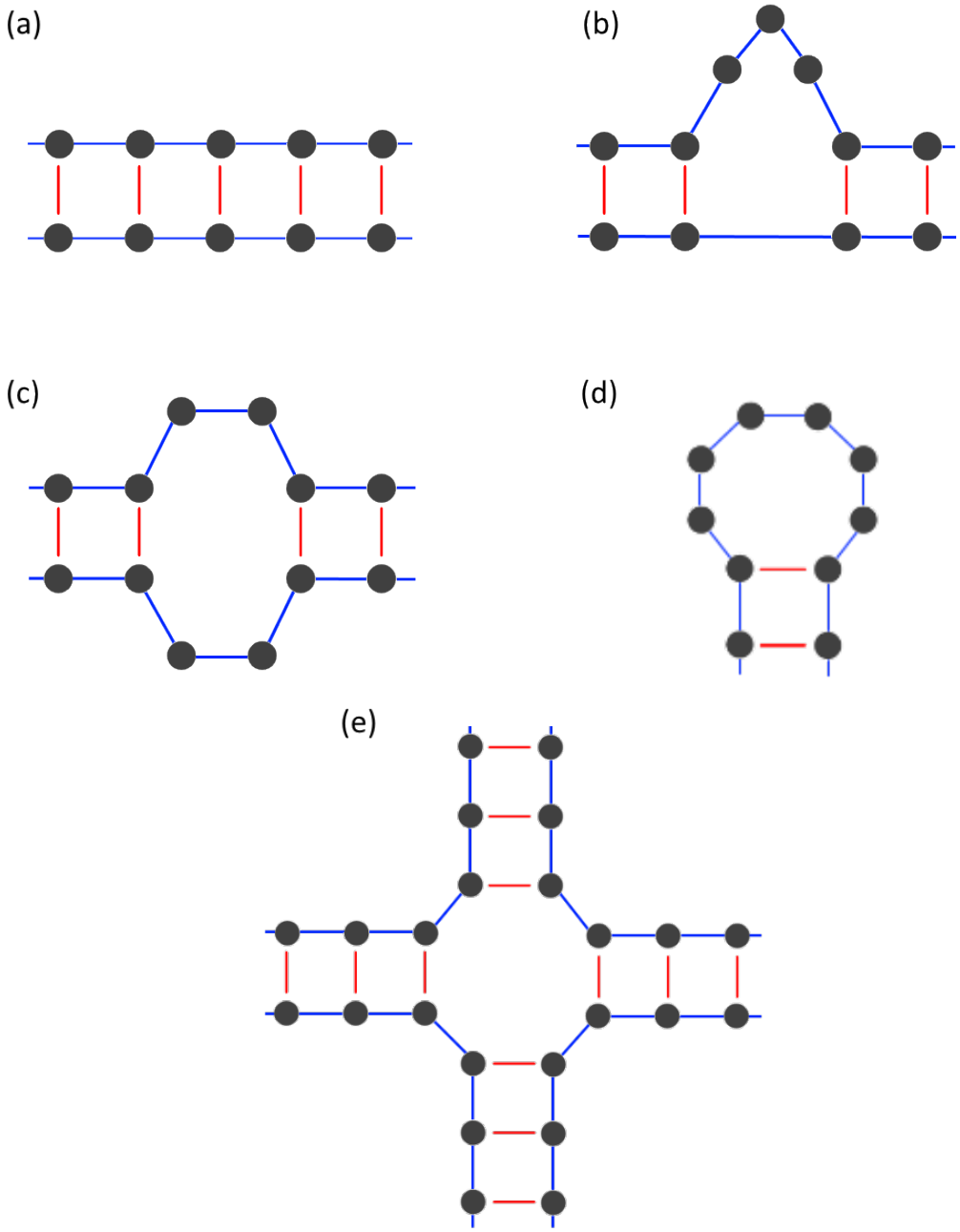

Figure 1.1: Basic structural elements of an RNA: (a) shows a helix, (b) a bulge, (c) an internal loop, (d) a hairpin loop, (e) a junction connecting 4 helices (i.e. a 4-way junction).

Figure 1.2: Pseudoknot-free base pair interactions.

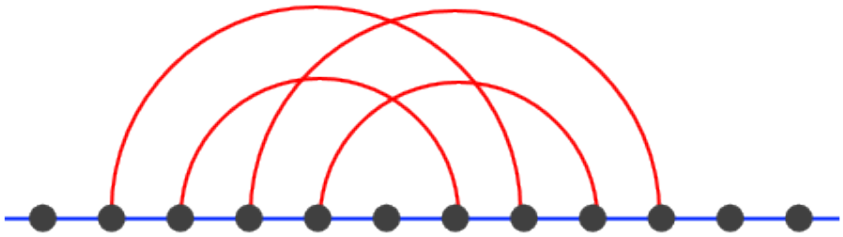

Figure 1.3: Pseudoknotted base pair interactions.

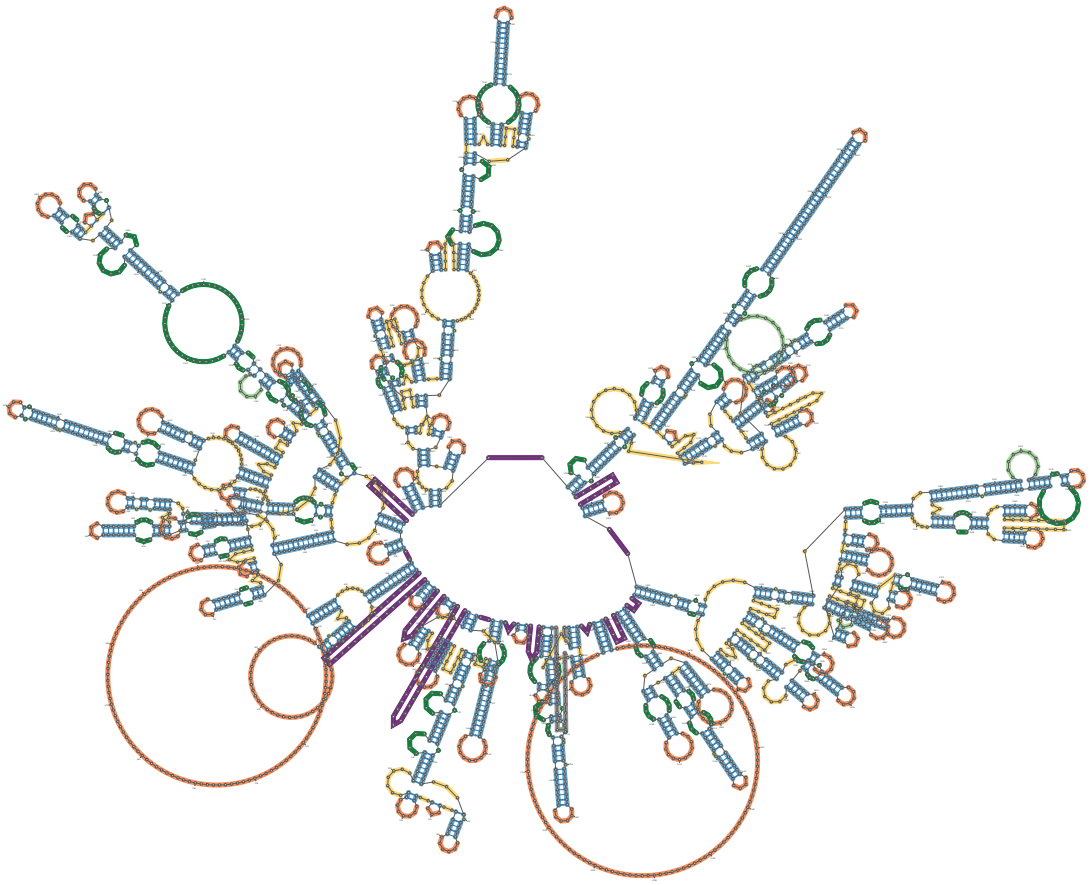

Figure 1.4: Secondary structure of a ribosomal RNA consisting of the basic structural elements, secondary structure illustration from the bpRNA database [37], RNA data from the Rfam database [76, 77].

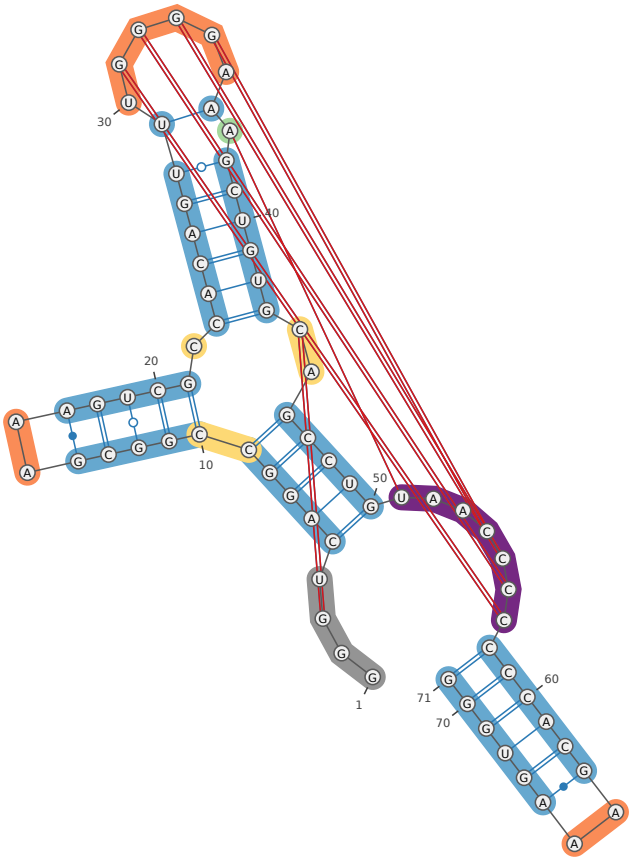

Figure 1.5: Secondary structure of a Zika virus ssRNA involving pseudoknots (depicted by the red lines), secondary structure illustration from the bpRNA database [37], RNA from the RCSB Protein Data Bank [120].

Different RNA secondary structures may indicate different "functional capabilities" of the RNA [14]. For instance, 4-way junctions are known to be involved in the functions of "hairpin ribozymes" [14,152], while 3-way junctions are formed and used by the "telomerase RNA" of "budding yeast species" [19]. In the latter case, the sequences were found to vary among species, but the 3-way junction structure is conserved [19], which suggests that secondary structures may be more important than the sequences themselves when it comes to enabling the RNAs to function and take part in catalysis. Alphaviruses, which are made of RNA, were also found to take advantage of specific secondary structures in order to affect recognition of foreign RNAs by the hosts they infect [65]. When it comes to the SARS-CoV-2 virus, although the functions are still unclear, parts of it were found to share similar secondary structures with other viral species related to SARS in the coronavirus family [118, 126]; which suggests that their close relationship could be inferred from the secondary structures. Since secondary structures are involved in various RNA functions, and as secondary structures may be conserved despite changes in the sequences; such structural information may also be useful for RNA identification purposes, in addition to the sequence information.

To summarize:

- Both sequence and secondary structure information are relevant to identify and classify RNAs.
- Specific sequence patterns may exist in RNAs whose functions depend on them, while the degree of sequence similarity imply the evolutionary distance (and therefore relationship) between the different sequences.
- However, secondary structures also play different roles in RNA functions; and in some cases, secondary structures and their functions are consistent between RNAs of different sequences.
- *Therefore, we hypothesized that using features representing the possible secondary structures that can be formed by a sequence of RNA, in addition to using sequence-based features with deep learning, would improve the overall RNA classification performance.*

Thus, our studies, which involve building secondary structure features and evaluating them with different datasets, sought to test and validate this hypothesis.

1.3 Problem Statements

We have tested our proposed approach in 2 different studies. Both studies involve RNA sequences in their datasets. The dataset in our first study is non-coding RNAs from the Rfam 14.1 database [76, 77] (see Section 4.1), while the second study involves RNA viruses from the NCBI Virus [18] (see Section 4.2). The problem statements for the first and second study are respectively as follow:

1. For a sequence of RNA belonging to one of the included Rfam [76, 77] classes, the classifier should correctly identify the corresponding RNA class using the provided features representing the RNA sequence (i.e. 1 class per RNA sequence) [134].

2. For a sequence of a RNA virus that is known to infect a specific host species, the classifier should correctly identify the susceptible host species among the other included species, given the features representing the viral RNA sequence [135].

1.4 Contributions

The following list outlines our contributions:

- Designed and developed a novel pipeline to produce "fingerprints" of RNA secondary structures from their sequences (consisting of scores based on free energy values) for use with deep learning, given a set of matching rules; [134];

- Leveraged an existing method to represent RNA secondary structures based on "graph theory" [44, 50, 68, 101] in order to create a set of matching rules to produce the RNA secondary structure fingerprints [134];

- Curated a set of common secondary structure motifs based on existing databases [37, 76, 77] to create an additional separate set of matching rules to produce a separate set of secondary structure fingerprints;

- Used and evaluated the secondary structure fingerprints and sequence-based features (k-mers with different values of k), both on their own and together, on the Rfam dataset [76, 77] with deep learning to classify RNAs into their Rfam [76, 77] classes [134];

- Finally, used and evaluated the secondary structure fingerprints and different sequence features (regular k-mers with varying k, and non-continuous k-mers or skip-mers with various configurations), both separately and combined, on datasets consisting of RNA virus sequences and the hosts they infect from NCBI Virus [18] with deep learning in order to predict susceptibility of the different host species given viral sequences [135].

The contributions and results of our work were presented in 2 accepted papers. They are as follows:

- "Assessing the Use of Secondary Structure Fingerprints and Deep Learning to Classify RNA Sequences" (2020) [134]
- "Extracting and Evaluating Features from RNA Virus Sequences to Predict Host Species Susceptibility Using Deep Learning" (in press) [135]

Chapter 2

Literature Review

Prior work on RNA classification and identification have used either features derived from sequence information, or features derived from secondary structure information. This chapter begins with discussions of these related work and their possible limitations. Then, the chapter continues with background information on a secondary structure representation technique, used as a basis of one of our secondary structure fingerprint sets. Finally, the chapter concludes with a discussion on deep learning.

2.1 Sequence Information for RNA Identification and Classification

2.1.1 K-mer Overview

Numerous bioinformatics studies involving sequences, both DNA and RNA sequences, have relied on k-mer instead of the raw sequences themselves. K-mer is also referred to as "n-grams" [82], which counts and measures the prevalence of different possible substrings from strings. In the context of bioinformatics, the possible characters in the substrings are restricted to letters that represent biologically relevant information – for instance, for a basic representation of a sequence of RNA, the possible characters would be "A", "C", "G", and "U". A more comprehensive k-mer representation may include more possible characters from the IUPAC code of nucleic acids, which may be used in an extracted RNA sequence when the exact nucleotides at the corresponding positions could not be determined [73].

K-mers of two different strings allow approximate measurement of their "distance and similarity" between each other [82]. K-mer forms a representation of a string by counting the occurrences of possible substrings of length k for a set of possible characters. As an example, for $k = 2$ with a set of possible characters consisting of "A", "C", "G" and "T", the possible substrings which occurrences in a string would be counted to form the k-mer representation of that string are: "AA", "AC", "AG", "AT", "CA", "CC", "CG", "CT", "GA", "GC", "GG", "GT", "TA", "TC", "TG", and "TT". The different distributions of these substrings in different sequences can then be used to estimate their similarity or dissimilarity [40]. As k-mer only counts occurrences of the possible substrings in a sequence, positional information of the substrings within the sequence is not retained. This also means that two different sequences may have the exact same k-mer representation if the occurrence counts of the possible substrings are the same, regardless of whether or not there are differences in positions of substring occurrences between the sequences [13].

In addition, depending on the length of the sequences, performing analyses and comparison on k-mers instead of the sequences themselves may be more efficient, especially for longer sequences. K-mers of length k with a number of possible substring characters of p will produce p^k features that can then be analyzed and compared with, irrespective of the sequence length. Thus, given the same amount of computational resources, analyzing k-mers instead of the original sequences will allow longer sequences to be included in the analyses.

2.1.2 K-mer in Prior Studies

K-mers have been used for various RNA identification and classification purposes in previous studies. For instance, k-mer was used in order to classify sequences of microRNA into their species [157]. In addition to sequence motifs, k-mer representations of "$k = 1, 2, 3$" were combined and used with random forest in the study, and an average classification accuracy exceeding 80% was achieved [157].

A different application of k-mers was demonstrated in another study, in which k-mer features were used to identify the functions of long non-coding RNAs (lncRNA) [81]. The authors hypothesized that using k-mers would be more useful to identify lncRNAs compared to checking for similarity using traditional alignment-based methods, as the existence of protein binding sequence motifs "may be more important" than their location in the lncRNA strand [81]. In other words, lncRNAs that are related in function could have similar sequence motifs without sharing similar complete sequences [81]. In addition, these binding motifs are often short, and therefore, the authors justified that k-mers with k

within the range of 3 to 8 may be useful to infer the prevalence of the short binding sequence motifs [81]. They indeed found and concluded that their k-mer based approach can be used to identify functionally related lncRNAs, including lncRNAs without "linear sequence similarity" [81].

When it comes to RNA viruses, k-mers have also been utilized to analyze and classify viral sequences. In one study, k-mer features were utilized with different machine learning algorithms (support vector machines, random forest, etc.) in order to classify sequences of HIV-1 viruses into their subtypes, which is a clinically relevant application [130]. By using and trying out different machine learning techniques to analyze k-mer representations of the different HIV-1 viral sequences, they achieved a maximum accuracy of 96.49% [130]. Furthermore, their k-mer based approach has also been tested with other species of RNA viruses, namely "dengue", "hepatitis B", "hepatitis C", and "influenza A"; and achieved overall classification accuracy that exceeds 95% [130].

A different study, which is related to one of our studies involving RNA viruses, has used k-mers as one of the input features in order to classify virus infectivity across 9 host genera [158]. They found that their "relative word frequency vector" performed best with random forest, which achieved area under the receiver-operating curve (AUC) of over 0.85 [158].

Similarly, another study also related to our RNA virus study has utilized k-mers to identify which host does a specific viral sequence infects; although unlike our study, the hosts are limited to prokaryotic hosts [3]. The authors claimed that k-mers are suitable for this problem; because viral replication relies on the "translational machinery of its host", and in turn, viruses may "share highly similar patterns in codon usage or short nucleotide words (k-mers) with their hosts" due to selection pressures [3]. With k-mer of length 9, one of their approaches covered in the study achieved an AUC of 0.90 [3]. However, when it comes to accuracy, only host prediction at the class, phylum, and domain levels managed to exceed 70% with the same length of k-mer [3].

Finally, another k-mer based application, VirFinder, has demonstrated that k-mer representations can be used to distinguish viral sequences from host sequences given "mixed metagenomes containing both viral and host contigs" [119]. The performance was measured by AUC as well, which exceeded 90% [119].

As k-mer features were found to be versatile in previous studies for a wide variety of classification and identification purposes, our studies have also included k-mers as one of the feature types for classification. K-mers were used both alone and in conjunction with other feature types, in order to compare their performance alone and the additional contributions k-mers make when combined with other features.

2.1.3 Non-continuous K-mer Overview

Despite the versatility of k-mers, the length of substrings that can be captured is limited to the value of k. If the sequence motif in a particular dataset that are useful for sequence identification is longer than k, the existence of these motifs relative to other sequence motifs will not be effectively represented in the k-mer representations; and as a result, the identification accuracy will be sub-optimal.

One solution to address this problem is by simply increasing the value of k. However, the size of the k-mer representation grows exponentially with every increase of k; since for k-mer with 4 possible nucleotides (4-mer), a representation for a single sequence consists of 4^k values. This, in turn, will result in a higher computational resources requirement needed to analyze and compare the k-mer representations with one another.

Another possible solution to cover longer sequence motifs is by using non-continuous k-mers, also known as "gapped k-mers" [23, 51], "skip-mers" [31], or "spaced k-mers" [13] in previous studies. Unlike regular k-mers, which substrings consist of characters that have to be matched against at all positions, substrings of non-continuous k-mers contains "gaps" [51] at certain positions which do not have to be matched against. In other words, an occurrence of a regular k-mer substring within a sequence match all of the substring characters at all positions, while an occurrence of a non-continuous k-mer substring within a sequence allow non-matching characters as specified by the non-continuous k-mer substring/configuration.

To illustrate this difference, consider the following example. One of the possible substrings of a regular 2-mer which set of possible characters are "A", "C", "G", "U" is "AU". The sequence "AAUAAU" has 2 occurrences of "AU" (at position 2 to 3, and at position 5 to 6). In contrast to the regular k-mer substring, an example of a non-continuous k-mer substring for the same set of possible characters is "A*U", where "*" represent a character that can be skipped (or matched with any character) at that position. Thus, the same sequence "AAUAAU" has 2 occurrences of "A*U", both of which are "AAU" (at position 1 to 3, and 4 to 6).

Due to the skippable characters, the number of possible substrings of a non-continuous k-mer is less than the number of possible substrings of a regular k-mer of the same substring length and same set of possible characters. As an example to illustrate this, a non-continuous k-mer which substring length is 3, skips every other character, and which possible characters are "A", "C", "G", and "U", will only have the following combination of possible substrings: "A*A", "A*C", "A*G", "A*U", "C*A", "C*C", "C*G", "C*U", "G*A", "G*C", "G*G", "G*U", "U*A", "U*C", "U*G", and "U*U", (i.e. 4^2 possible substrings, where 2 is the number of characters that have to match); whereas a regular k-mer

with the same set of possible characters and same substring length will have 4^3 possible substrings. These "gaps" [51] or skips allow non-continuous k-mers to capture prevalence of longer sequence motifs with the same number of possible substrings combination, which also means that the resulting representation to be analyzed and compared will also be of the same size (i.e. have the same number of values representing the counts of each possible substrings) as a regular k-mer representation (of equivalent substring length and the same set of possible characters) that can only capture shorter sequence motifs. Despite of this advantage, it is important to note that due to the inexact nature of the matches, non-continuous k-mers are less suitable to capture exact sequence motifs, and are only capable of capturing inexact ones.

Different forms of non-continuous k-mers specification have been used in prior studies. For a simple non-continuous k-mer used in our study [135] and in a past study [31], consecutive characters that have to match are followed by the consecutive skipped characters, and vice versa – in short, consecutive match characters and consecutive skip characters alternate.

In addition to the set of possible characters that is also applicable to regular k-mers, a configuration of a simple non-continuous k-mer can be defined by the following [31, 135]:

- Number of consecutive characters that have to match,
- Number of consecutive characters that do not have to match (i.e. skipped characters),
- Either length of the substrings (i.e. sum of the number of matching characters, and the number of skipped characters) [135], or the number of matching characters [31].

An example configuration that illustrates how the three specification can define non-continuous k-mers is shown in Figure 2.1.

In addition to this setup of non-continuous k-mers, other studies have also experimented with more specific non-continuous k-mers, in which the positions of matching characters and skipped characters are predetermined instead of alternating [13]. A different, more flexible setup of non-continuous k-mers allow the skipped/mismatching characters to be placed in any positions within the substrings, as long as the number of mismatching characters is equal to the expected number of skipped characters [51].

2.1.4 Non-continuous K-mer in Prior Studies

The effectiveness of regular k-mers and "gapped" k-mers as features to distinguish different sequences were compared in one study that attempts to classify "Transcription Factor

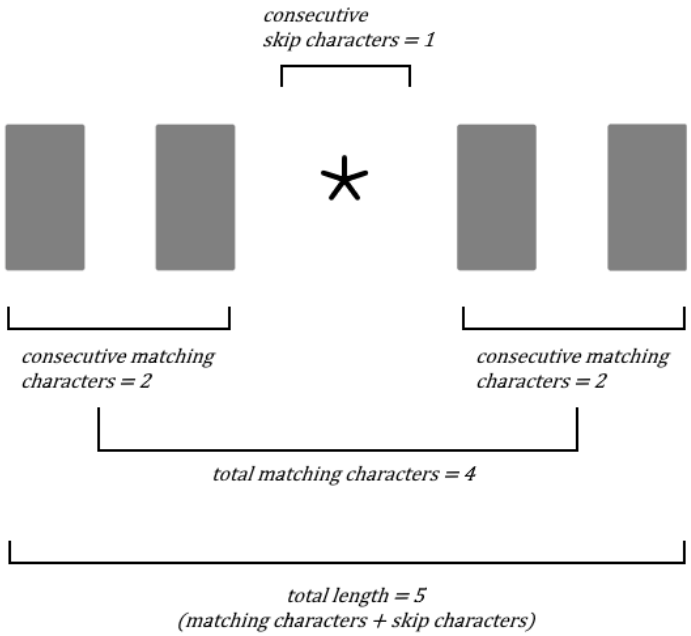

Figure 2.1: An example configuration of non-continuous k-mer with alternating matching and skipped characters. Boxes indicate positions of characters that have to match, while asterisk indicate the position that can be skipped. Adapted from [31].

Binding Sites (TFBS)" [51]. The different types of k-mers were used separately with support vector machine in order to predict "CTCF" [100] "binding sites" [51]. "CTCF" was chosen due to its known capability to bind to long sequences [51], while the human genome was used as the "binding specificity" of the transcription factor has previously been analyzed in the genome [51, 80]. For regular k-mers in the study, different values of k from 6 to 20 were used and assessed; while for the gapped counterparts, the number of matching characters remained at 6, but the different total length values ranging from 6 to 20 (with varying/flexible gap positions instead of alternating) were assessed [51]. The prediction performance for the tested representations was measured in AUC. As expected by the authors, due to the known long "binding sites" sequences, gapped k-mer of total length 14 with 6 matching characters performed best with an AUC of 96.7%, whereas the best result of regular k-mers was achieved with $k = 10$ resulting in a lower AUC of 91.2% [51]. In addition, the authors noticed an interesting trend where using k-mers with $k > 10$ resulted in overfitting, whereas their gapped k-mers did not result in overfitting

even when the total substring length is increased [51].

Another study has compared k-mer, alternating "spaced k-mer" (where consecutive matching characters are followed by consecutive skipped characters, and the latter is then again followed by the former), and "irregular" "spaced k-mer", when it comes to rebuilding the phylogenetic trees – trees that depicts the evolutionary relationship between the different sequences [15]. Traditional methods to build such trees involve the computationally demanding sequence alignment process, which means similarity in different sequences are aligned together first before comparison is made. This process is needed as changes from one sequence to another may involve insertions or deletions, thus the similar and relevant parts of the different sequences may be located in different positions. However, none of the k-mer based method require sequence alignment [15]. The different types of k-mers were used to estimate how close or how far are 2 different sequences related to each other, and thus, the tree depicting the relationships between different sequences can be built from these estimated relatedness. Both DNA and protein sequences, and both real and simulated sequences, were used in their study [15]. The performance measurements between the different types of k-mers are made by comparing the trees rebuilt from the different k-mer features with the corresponding "reliable" reference trees [15]. To summarize their findings, when it comes to synthetic DNA sequences, they found that the spaced k-mers performed better than the continuous k-mer; with the number of matching characters of 10 for the former, and $k = 10$ for the latter (therefore the numbers of characters that have to match for both types of k-mers are equal; i.e. 10 was chosen as $k = 10$ performed the best with the continuous regular k-mer) [15]. However, when using a different DNA dataset, namely the "primate mitochondrial genomes", they found that using the regular continuous k-mer (with $k = 8$, as it was the best performing value of k when used with the regular k-mer) resulted in a more accurate rebuilt tree, compared to when the tree is rebuilt with the spaced k-mers (with the number of matching characters of 8, in order to match $k = 8$ as done with the previous dataset). When it comes to the datasets containing protein sequences, $k = 4$ performed best in all but one of the datasets. Therefore, similar to the approach used to test the different k-mers with the DNA sequence datasets, the number of matching characters used for the spaced k-mers are 4 as well. The results show that, for all of the real datasets and almost all of the synthetic datasets, all but one of spaced k-mers consistently outperformed the continuous k-mers [15]. Unlike the previous study which results conclusively show that non-continuous k-mers performed better than the regular continuous k-mers, the results of this study implies that the performance of the different types of k-mers depends on the dataset being studied.

Similar to continuous k-mers, non-continuous k-mers can be used for a variety of applications, as shown by previous study solving different problems. However, between the

two types of k-mers, one type may be more suitable than the other type depending on the dataset and the problem being solved. Therefore, as with the other feature types used in our study; to measure how non-continuous k-mers perform on their own compared to the other feature types, and how non-continuous k-mers affect classification results when combined with the other feature types; classification performance using both non-continuous k-mers alone, and non-continuous k-mers combined with the other feature types, were evaluated in our study.

2.2 Secondary Structure Information for RNA Identification and Classification

Due to the importance of secondary structure in functional RNAs [140], secondary structure information may provide useful information about the identity or class of a sequence of RNA. This especially true for RNAs that do not code for proteins, but do have their own specific functions; as structural information of such non-coding RNA are known to provide clues about their different functions [46, 99].

Different tools to either predict or extract secondary structure information from a sequence exist, such as IPknot [122] and RNAMotif [96]. Depending on the identification/classification approach, these tools are usually used to derive the structural information, which is then used in order to identify the RNA in question.

2.2.1 Secondary Structure Information in Prior Studies

In one study on RNA classification, the IPknot program [122] was used in order to turn RNA sequences into predicted secondary structures, after which the predicted secondary structures are broken down into their substructures with a different program named MoSS [16], which are then converted into deep learning input vectors [46]. A sequence may have a specific broken down substructure, whereas another sequence may not have that specific substructure; and therefore, the deep learning feature vectors in the study were formed by concatenating boolean values indicating whether or not a specific substructure can be found in the predicted secondary structure for each of the sequences [46]. The dataset was obtained from Rfam [76, 77], and 13 RNA classes as annotated in the Rfam dataset were used in the study: "miRNA, 5S rRNA, 5.8S rRNA, ribozymes, CD-box, HACA-box, scaRNA, tRNA, Intron gpI, Intron gpII, IRES, leader, riboswitch" [46]. A deep learning architecture consisting of convolutional neural network (CNN) layers was trained with their

training dataset, and the trained models were evaluated using 10-fold cross validation with a separate dataset which does not contain any of the sequences used during the training process [46]. This proposed approach of using broken down predicted secondary structures with a CNN-based deep neural network resulted in an overall classification accuracy of 74% [46].

A related study that also uses the IPknot [122] program proposed treating the predicted structures as graphs, then extract the "graph properties" of the different graphs, which are then used as features to different machine learning algorithms, e.g. "BayeNet", naive bayes, multilayer perceptron, random forest, support vector machine, sequential minimal optimization, "IBk" classifier [113]. The "graph properties" extracted from the graphs representing the predicted RNA secondary structures are as follows: "articulation points", "average bibliographic coupling", "average Burt's constraint", "average closeness centrality", "average co-citation coupling", "average coreness", "average degree". "average edge betweenness", "average node betweenness", "average path length", "diameter", "girth", "graph density", "maximum coreness", "transitivity", "variance of Burt's constraint", "variance of closeness centrality", "variance of coreness", "variance of edge betweenness", and "variance of node betweenness" [113]. Two sets of values representing these graph properties were used with the machine learning algorithms: the first set consists of the raw values, and the second set consists of values normalized to between -1 and 1 [113]. Finally, Rfam [76,77] was also the source of the dataset used in this study, and 18 of the RNA classes were used [113]. This classification approach resulted in a maximum sensitivity of 43.3% (using RandomForest with the raw instead of normalized values of the graph properties) [113].

Finally, another attempt at using graph properties for RNA classification was made by a similar study [28]. The Rfam [76, 77] dataset was used as well to test the approach, but one difference between this and the previous study is the program used to predict the RNA secondary structures from the RNA sequences – the RNAfold program belonging to ViennaRNA [61] predicted the secondary structures [28]. Using graph properties alone with support vector machine to perform the classification, an overall "Matthew's correlation coefficient" (MCC) of 0.32 was achieved [28]. However, when the graph properties were combined with other features containing "sequence homology" information, the overall MCC increased to 0.446 [28].

2.2.2 Possible Limitation of Prior Studies

One common theme among these prior studies is how the initial secondary structure information is obtained – that is, by using a single predicted secondary structure produced

by a prediction tool for each single sequence. Information about other possible secondary structures that can be formed by the same RNA sequence, either of equal or different thermodynamic stability (i.e. free energy of the folded structure), is therefore not obtained in this initial step.

However, it is possible for the additional information on other possible secondary structures to contribute to the classification/identification performance if such information was obtained and used; as RNAs are indeed known to be able to form different structures, some RNA species can stop folding at "non-equilibrium states" [140], and the actual RNA folding processes are not always going towards the least free energy (thermodynamically stable) structure [134,140]. In addition, as previously mentioned, RNA secondary structure formation processes *in vivo* may be affected by "RNA chaperone", which are proteins that are capable to influence and guide the RNA folding processes; which means that the actual formed secondary structures *in vivo* may differ from the secondary structures formed *in vitro* or the secondary structures predicted *in silico* [59, 134]. Using a single predicted secondary structure for each single sequence does not account for these extra possibilities.

Furthermore, it is also possible for the single predicted secondary structure of a single sequence to be inaccurate, or for a single sequence to have higher free energy (less thermodynamically stable) alternative secondary structures that "just as likely to be correct as the minimum free energy one"; due to "errors in the parameters and assumptions in the calculation" [140].

This suggests that there is a possibility classification performance may be limited when relying only on a single predicted secondary structure for each sequence. In other words, taking into account the other possible secondary structure conformations instead of using only one secondary structure for a sequence may provide additional useful information to the classifier (e.g. machine learning algorithms or deep learning), which in turn, may result in a greater overall classification or identification performance.

2.3 Graph Representation of Secondary Structure

Graph representations can be used to simplify complex secondary structure information, and as an extension, a collection of graph representations can be used to represent a collection of possible secondary structures that can be formed by a strand of RNA. The collection of graph representations can be leveraged to represent a collection of possible secondary structures for a single sequence. This provides a possible avenue to address the limitation of prior related RNA classification approaches that use single predicted

structure information per sequence. However, in order for the representation to be useful, only biologically relevant graph representations should be used; using an arbitrary graph representation or an arbitrary method to convert secondary structure information into graphs may introduce biologically irrelevant information to the classifier.

2.3.1 "RNA-As-Graphs"

One of the existing methods to produce biologically relevant graph representations given RNA secondary structures is "RNA-As-Graphs" (RAG) [44, 50, 68, 101]. It was designed based on the "graph theory", in which a single graph represents a secondary structure motif, i.e. a pattern in the structure that may be shared by different RNAs [50]. The different graphs allow quantitative classification of the secondary structure motifs by their "topological properties" and "connectivity patterns" [50, 134]. Hypothetically possible structural motifs, in addition to motifs based on existing known RNA secondary structures, are included in their collection – the inclusion of the former type is to aid research and discovery of novel RNA secondary structures [50, 134]. The collection of these graph representations are available from the RAG database [50].

Two different types of graphs are used in RAG: tree graphs, and dual graphs [50]. The tree graphs are able to convey structural motif information containing any RNA secondary structure elements (e.g. bulges, loops, etc.) with the exception of pseudoknots, while the more complex dual graphs are additionally capable of conveying pseudoknots [50]. Due to this difference, the steps to produce a graph motif representation from the raw secondary structure also differ between the two types of graphs.

A tree graph structural motif representation is produced as follows [50] (illustrated in Figure 2.2):

1. A bulge, hairpin loop, or internal loop with at least 2 unpaired nucleotides is converted into a vertex [50],
2. The ends of the RNA secondary structure (5' and 3' ends) are also turned into a vertex,
3. A stem/helix with at least 2 paired nucleotides is converted into an edge,
4. And finally, a junction with 3 or more stems/helices forms a vertex.

On the other hand, the following are the steps to produce a dual graph representation from a secondary structure [50] (illustrated in Figure 2.3):

1. A stem/helix with 2 or more consecutive paired nucleotides becomes a vertex,

2. Any other secondary structure elements (such as a loop, a junction, a bulge with one or more unpaired nucleotide, and a stem that does not meet the criteria to be a vertex) is represented as an edge.

Regardless of the differences between the two types of graphs, the complexity of the represented structural motif can be measured by the count of vertices in the graph [50]. In addition, a graph representation with a lesser number of vertices may be a subgraph of graphs with more vertices.

In short, RAG provides a biologically relevant way to represent RNA secondary structure motifs, and the representation includes both known secondary structures and hypothetical secondary structures [50]. The graphs serve as high level topological representations, and as such, different RNA secondary structures may correspond to the same graph representation. Due to its biological relevance and applicability; our study leveraged this method to produce secondary structure fingerprints (derived from matches with the different graphs/motifs, as described in Section 3.1), which include information about not just one but multiple possible secondary structure conformations per strand of RNA [134].

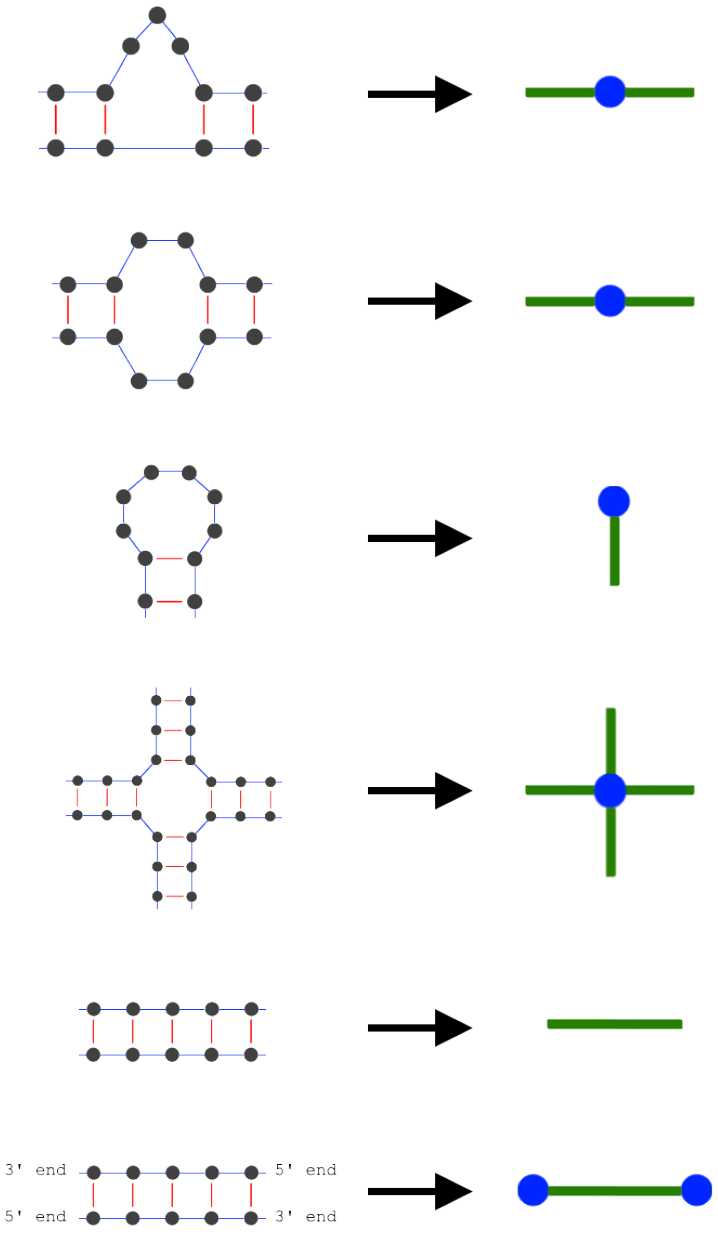

Figure 2.2: Different secondary structure elements and their tree graphs.

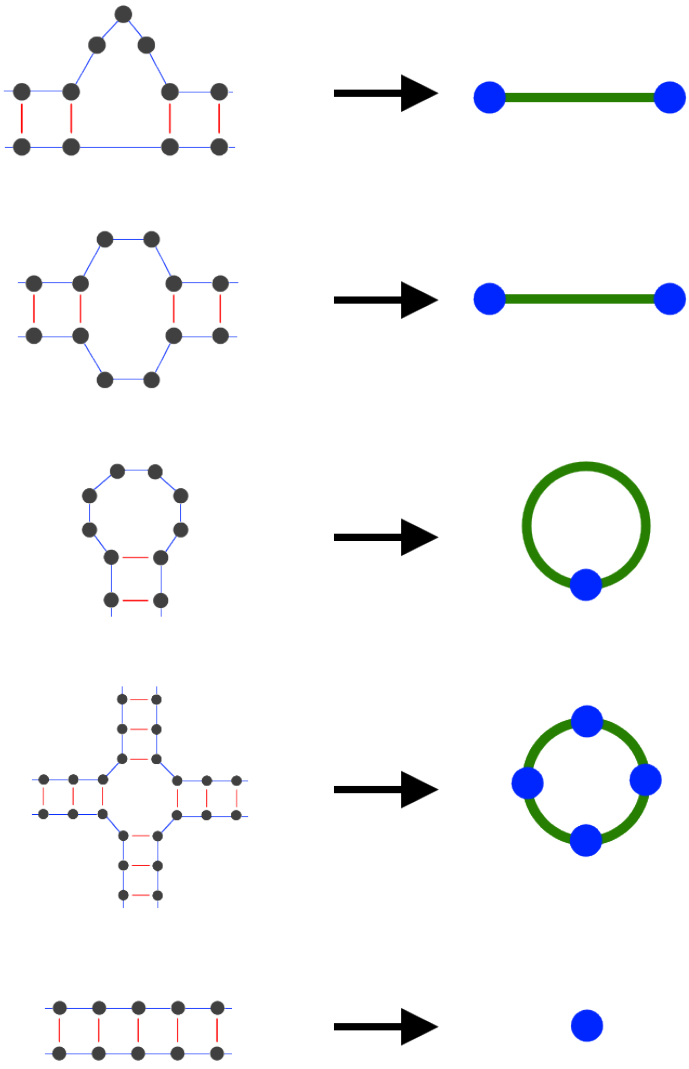

Figure 2.3: Different secondary structure elements and their dual graphs.

2.4 Deep Learning

2.4.1 Supervised Learning and Artificial Neural Network

Machine learning strives to create algorithms/methods that are capable to learn from experience [55]. Different approaches to machine learning are available [38]. In our studies, we focused on supervised learning with deep neural networks (which are artificial neural networks).

Given a model that takes an input vector, and produces an output by processing the input with a set of adjustable parameters; supervised learning refers to the process of adjusting these parameters using known inputs and outputs, such that the model well represents the relationship between the inputs and outputs [35]. Given enough learning capacity of the model, adequate learning steps with sufficient quantity and quality of input-output samples, a trained model would then be able to produce the correct output given a new input. For a classification problem, which is relevant to our studies (e.g. the type of RNA), the output is a representation of the class.

A "neuron", which is based on actual neurons in a brain, is such trainable model – as it takes input, processes the input with its parameters, and produces an output [38]. In addition to parameters that are adjusted during the learning process, a neuron also has parameters that are specified instead of learned (e.g. the "activation functions" [123]) – these are also known as "hyperparameters" [45].

An "artificial neural network" connects different neurons together [38], and the entire network itself also forms a trainable model as a whole; since it receives inputs, has parameters (consisting of the parameters held by individual neurons and the connections between the neurons), and outputs processed results. A neuron within a network can receive input values from outside of the neural network (such neurons that receive inputs from the outside are also known as "input neurons"), or from another neuron within the neural network (i.e. an output value of another neuron is used as an input) [123].

How the connections are made between the different neurons determine the shape of the neural network, the flow of values, and therefore its overall behaviour and capability. The connections between different neurons may also be "weighted", which means that the outgoing value to a neuron may be changed and differ from the incoming value [123]. These weights can be adjusted as well during the learning process [123].

Supervised learning for a neural network works by providing an input, checking its output, comparing the output with the desired output, and backpropagating the values

derived from this difference between the outputs in order to adjust the weights (i.e. the adjustable parameters); such that next time the neural network receives the input, the output produced by the adjusted weights would be closer to the expected output [123]. The backpropagated values are determined by and thus dependent on the "loss function" [36]. This process is typically repeated as needed (e.g. until a specified condition is met), or up to a specified number of repetitions.

The consistency and quality of the samples used for training affects the outcome of supervised learning, specifically the consistency and quality of the relationship (even if complex) between the input features and the correct output [9].

In addition, it is also common for features to be preprocessed prior to being used, such as being "normalized" [67, 147], so that the possibly different ranges of raw values will not introduce additional inconsistency.

2.4.2 Deep Neural Network

Neurons in a neural network are typically arranged in layers; with an input layer consisting of neurons that receive input from outside of the network, hidden layers with neurons that process/transform the input (using their parameters), and an output layer with neurons that output the resulting processed data [38]. With this arrangement, the neurons on each layer are connected to the neurons belonging to the next layer.

The number of layers between the input layer and the output layer determines whether the network is "shallow" or "deep" – the latter is used in deep learning applications [123]. Deeper networks with more layers and neurons have more parameters that can be adjusted during training, which provide the neural network with greater learning capacity. This makes deep neural networks more capable to handle a wider variety of problems than the shallow neural networks, and even outperform other various machine learning algorithms [20, 123, 144, 145].

Although more neurons, more connections, and/or deeper layers allow the neural network to have greater capacity to learn, creating a neural network with too many neurons, connections and/or layers can result in "overfitting" [36] – that is, the neural network will be overly adapted to the data used to train it to the point where the ability to produce correct outputs from unseen data is diminished. In other words, the trained parameters (e.g. weights) do not represent a generalized solution for the actual problem to solve, but are only specific to the trained data. Therefore, the size of the neural network should be decided depending on the nature of the problem and the dataset.

In our studies, different types of features derived from the RNA sequences have been used, separately and combined, with supervised learning of deep neural networks to solve the same problems; which allowed us to assess the overall quality, performance, and how informative is each set of the different feature types when it comes to solving the problem (i.e. RNA classification) compared to others [134, 135].

2.4.3 Hyperparameters

The parameters involved in a deep learning process can be categorized into two types: learned parameters [125] and hyperparameters [45]. Learned parameters, as the name suggests, refer to parameters that are determined automatically during the learning process, such as the weights of the connections between neurons [125]. In contrast, hyperparameters refer to parameters about the model that are determined prior to the learning process, which include but are not limited to the previously discussed "number of layers" and "number of units" per layer – these parameters affect the learning process and therefore would affect the learned parameters at the end of learning [45].

The optimal hyperparameters can be determined manually, or by using "hyperparameter optimization" algorithms [45]. However, when it comes to deep learning, especially with larger models and datasets, such automated hyperparameter optimization is computationally demanding – it requires large amounts of computational resources at the time of this writing, making it "a hard problem in practice" [45]. Due to this computational constraint at the present, and as automated hyperparameter search is not the focus of this thesis, manual search was the primary method used to determine the hyperparameters of the deep learning models in our studies.

Several key hyperparameters that are relevant to our studies include:

- *"Number of layers"* [45]: depth of the model,
- *Number of neurons on each layer* [45]: width of each layer,
- *Number of "epochs"* [75]: how many times/repetitions should the model be trained with the entire training dataset,
- *"Batch size"* [75]: how many training entries should be used to train and adjust the model parameters at a time,
- *"Learning rate"* [29]: the degree of adjustments that can be made to the trained parameters at a time during the training process,

- *"Learning rate schedule"* [93, 136]: rules allowing the use of different/dynamic learning rates at different phases of the training process,
- *"Optimizer"* [29]: algorithm consisting of "update rules" – that is, how should the trained parameters be updated given the currently produced output and the expected output during training,
- *"Loss function"* [72]: a function that produces values representing the differences between the model-predicted and expected output.

2.4.4 Classification Performance Measurement

There are several metrics that can be used to measure the classification performance of classifiers (i.e. how correct are the classifications), including deep learning based classifiers. These metrics take the following counts into account in order to calculate the classification performance:

- *True positive*: number of items that are correctly predicted as belonging to a specific class/category,
- *False positive*: number of items that are *incorrectly* predicted as belonging to a specific class/category (i.e. the items are actually of a different class/category),
- *True negative*: number of items that are correctly predicted as *not* belonging to a specific class/category, and
- *False negative*: number of items that are *incorrectly* predicted as *not* belonging to a specific class/category (i.e. the items are actually of that specific class/category).

The metrics and their methods of calculation using these counts are summarized in Table 2.1 [115].

For a binary classification problem (i.e. for each input, there are only 2 possible output classes), these formula work well. However, for "multi-class classification" [129] problems, where there are more than 2 possible classes to classify the inputs into, further calculation steps need to be taken to obtain the overall classification performance covering all of the classes.

The aforementioned counts used by the formulas (true positive, false positive, true negative, and false negative) only considers the classification of 1 class (i.e. *positive*)

Table 2.1: Different metrics to measure classification performance.

Metric	Calculation Formula [115]
Accuracy	$\frac{true\ positive\ +\ true\ negative}{true\ positive\ +\ false\ positive\ +\ true\ negative\ +\ false\ negative}$
Recall/Sensitivity	$\frac{true\ positive}{true\ positive\ +\ false\ negative}$
Precision	$\frac{true\ positive}{true\ positive\ +\ false\ positive}$
F1	$\frac{true\ positive}{true\ positive\ +\ \frac{false\ negative\ +\ false\ positive}{2}}$

relative to one or more other classes (i.e. *negative*). Thus, in a multi-class problem, these formulas will only allow performance classification of a single class at a time (i.e. 1 class is considered the *positive* class, whereas the rest of the classes is considered the *negative* class). In order to calculate the overall performance of the classifier spanning across all of the classes; we can calculate the metrics for each of the classes (relative to the rest of the classes) at a time, then take the average of these. For example, to calculate overall accuracy of a classifier when it comes to classifying all of the available class, the accuracy values for each of the classes would be calculated, these values would then be added/summed together, then the sum can be divided by the number of classes [129]. This is illustrated in (2.1), (2.2), and (2.3) for accuracy, recall/sensitivity, and precision respectively, where m represents the number of classes and c represents the current class [129].

$$macro\ avg.\ accuracy\ =\ \frac{\sum_{c=1}^{m} \frac{true\ positive_c\ +\ true\ negative_c}{true\ positive_c\ +\ true\ negative_c\ +\ false\ positive_c\ +\ false\ negative_c}}{m} \tag{2.1}$$

$$macro\ average\ recall\ =\ \frac{\sum_{c=1}^{m} \frac{true\ positive_c}{true\ positive_c\ +\ false\ negative_c}}{m} \tag{2.2}$$

$$macro\ average\ precision\ =\ \frac{\sum_{c=1}^{m} \frac{true\ positive_c}{true\ positive_c\ +\ false\ positive_c}}{m} \tag{2.3}$$

However, these formula do not take any class imbalance into account. Only the number of classes is taken into account, meaning the average is "per-class" [129]. These are also known as *macro averages* (e.g. macro accuracy, macro recall, and macro precision).

Since our studies involve class imbalance, further adjustments are needed so that the measurements take such imbalance into account. *Micro* accuracy, recall, and precision take class imbalance into consideration [129]. The adjusted formulas for micro accuracy, micro recall, and micro precision are shown in (2.4), (2.5), (2.6) respectively. These are the metrics that we used in our studies.

$$micro\ avg.\ accuracy = \frac{\sum_{c=1}^{m} true\ positive_c + true\ negative_c}{\sum_{c=1}^{m} true\ positive_c + true\ negative_c + false\ positive_c + false\ negative_c} \tag{2.4}$$

$$micro\ average\ recall\ =\ \frac{\sum_{c=1}^{m} true\ positive_c}{\sum_{c=1}^{m} true\ positive_c\ +\ false\ negative_c} \tag{2.5}$$

$$micro\ average\ precision\ =\ \frac{\sum_{c=1}^{m} true\ positive_c}{\sum_{c=1}^{m} true\ positive_c\ +\ false\ positive_c} \tag{2.6}$$

In order to measure the generalized performance of a trained model, i.e. how well does the model represent the general relationship between the input values and the expected output values (as opposed to only knowing which output values to produce given a set of input values the model was trained with), the data used to evaluate the model needs to be separate from the data used to train the model [95, 116]. For this reason, it is common for machine learning classification related studies to "randomly split" the dataset into separate training set and evaluation/test set [95].

However, even with the separation of training and evaluation datasets, there may still be "sample representativeness issues" [95], which means that the evaluation results may still be specific to the specific training set and evaluation/test set used (i.e. the evaluation results may be different using a different training and evaluation sets). To address this issue, cross-validation techniques are often used. These techniques, which include the "k-fold cross-validation" and "leave-one-out cross-validation" (LOOCV), provide a more general assessment of the model performance which does not depend on a specific training set and a specific evaluation/test set [12]. Among the different techniques, LOOCV and 10-fold cross-validation were found to result in least bias [12, 103, 128]. As such, the studies covered by this thesis have employed the 10-fold cross-validation technique [12] in order to produce generalized performance assessments of the different deep learning models and deep learning features. Additionally, since the datasets in our studies involve class imbalances, the "stratified random sampling" technique [12] was also employed. This ensures that the distribution of classes are preserved between each of the 10 data subsets [12].

2.4.5 Deep Learning in Prior Studies

The capability of deep learning techniques to uncover unknown and complex relationships between the input values/features and output values has been shown in previous study, making deep learning suitable for solving problems where such relationship between the input and output is unknown [7, 134]. In cases of RNA classification and identification problems, this means that *a priori* knowledge about the relationships between the input features being used (e.g. secondary structure features, or k-mers) and the classes the RNA sequences are being classified into (e.g. the type of RNA) are not required, and can be discovered by the deep neural networks during the training process [7, 134].

Due to this robust capability, deep learning has been utilized by various published studies in bioinformatics [5,7,46,57,134]. The input features being used with deep learning and the problems to solve have also varied. For instance, "circDeep" uses 3 different types of "sequence descriptor" with deep learning in order to determine whether a sequence of RNA is a circular RNA or a different type of long non-coding RNA (i.e. a binary classification problem) [25]. Their network configuration includes the "asymmetric convolutional neural network (ACNN) and bidirectional long short-term memory (BLSTM)" architectures – the latter allows dependencies or relationships between different parts of their sequence representation to be captured [25]. With their approach, a maximum classification accuracy of 94.17% was achieved [25].

Another application of deep learning named "lncRNAnet" used preprocessed and encoded sequence information in order to predict whether or not a sequence is a long non-coding RNA [10]. Their application consists of "convolutional neural networks" (CNN) to obtain indicators of open reading frames (ORFs) in a sequence, and "long short-term memory (LSTM)" to determine if the sequence is a long non-coding RNA based on the obtained ORF indicators and the one-hot encoded sequence [10]. This approach attained classification accuracy between 81.20% to 91.83% depending on the dataset being used [10].

Finally, a prior study related to the topic of this thesis, which involves multi-class RNA classification from secondary structure information, had also utilized deep learning and the CNN architecture in order to learn the relationship between the existence of specific substructures in the predicted secondary structure and the class of RNA [46]. With their deep learning based approach, a classification accuracy of approximately 74% was obtained, which the authors had found to be superior to similar approaches using other machine learning techniques in place of deep learning (such as support vector machine, which only resulted in a lower overall classification accuracy of 67.36%) [46].

Chapter 3

Methods

The first two sections of this chapter cover our two types secondary structure fingerprints (section 3.1 and 3.2). Sequence-based features, namely k-mers and "skip-mers" [31], are then covered in the next two sections (section 3.3 and 3.4). Finally, section 3.5 covers the deep learning aspect involved in all of our studies.

3.1 Graph-Based Secondary Structure Fingerprints

One of the feature types used in our study to classify RNAs is the secondary structure fingerprints based on "RNA-As-Graphs" (RAG) [44, 50, 68, 101]. Several steps were involved in designing our approach to produce the RAG-based secondary structure fingerprints given an RNA sequence.

First, we decided to use the dual graphs representation instead of the tree graphs, as the former graphs are able to represent pseudoknots (which are known to have various roles in functional RNAs [17, 52, 84, 132]) in the secondary structures [44, 50, 134]. In other words, using the former graph type would allow more information to be represented in the resulting secondary structure fingerprints; as dual graphs representations also capture information that can be represented by the latter graph type [44, 50], but not vice versa.

We then decided to use dual graphs that are of 2 to 5 vertices. This limit is intended to create a balance between the resulting deep learning input feature size (which depends on the number of graphs) and the information contained by the fingerprints. Without the limit, the deep learning input feature size may be overly large; which in turn, would negatively impact the computational requirements of our proposed approach.

There are 3 possible dual graph representations with 2 vertices, 7 possible representations with 3 vertices (out of 8 possible graph combinations), 17 possible representations with 4 vertices

(out of 29 possible graph combinations), and 20 possible representations with 5 vertices (out of 110 possible graph combinations) [44, 50]. Thus, in total, there are 47 graphs that are included to build the secondary structure fingerprints.

Then, RNAMotif "descriptors" [96] representing these graphs are created. RNAMotif is a bioinformatics tool that, given a sequence and a descriptor specifying secondary structures to look for, searches for structural matches that can be formed by the provided sequence or parts of the sequence [96]. In addition, the program also has the capability to score the secondary structure matches that it finds [96]. Different scoring mechanisms, which users can choose or derive from, are built into the program [96].

The relationship between a descriptor, sequence, and structural matches with their scores in the context of RNAMotif is illustrated in Figure 3.1. Meanwhile, an example descriptor specifying a hairpin loop of 4 nucleotides in the loop and 3 nucleotides in the helix, is shown in Figure 3.2. For an example input sequence of "AAAAAAAAAAGGGGUUUUUUUUUUUU-UUUUUUUUCCCCAAAAAAAAAAAA", matches found and scored by RNAMotif using the previously mentioned example descriptor is depicted in Figure 3.3.

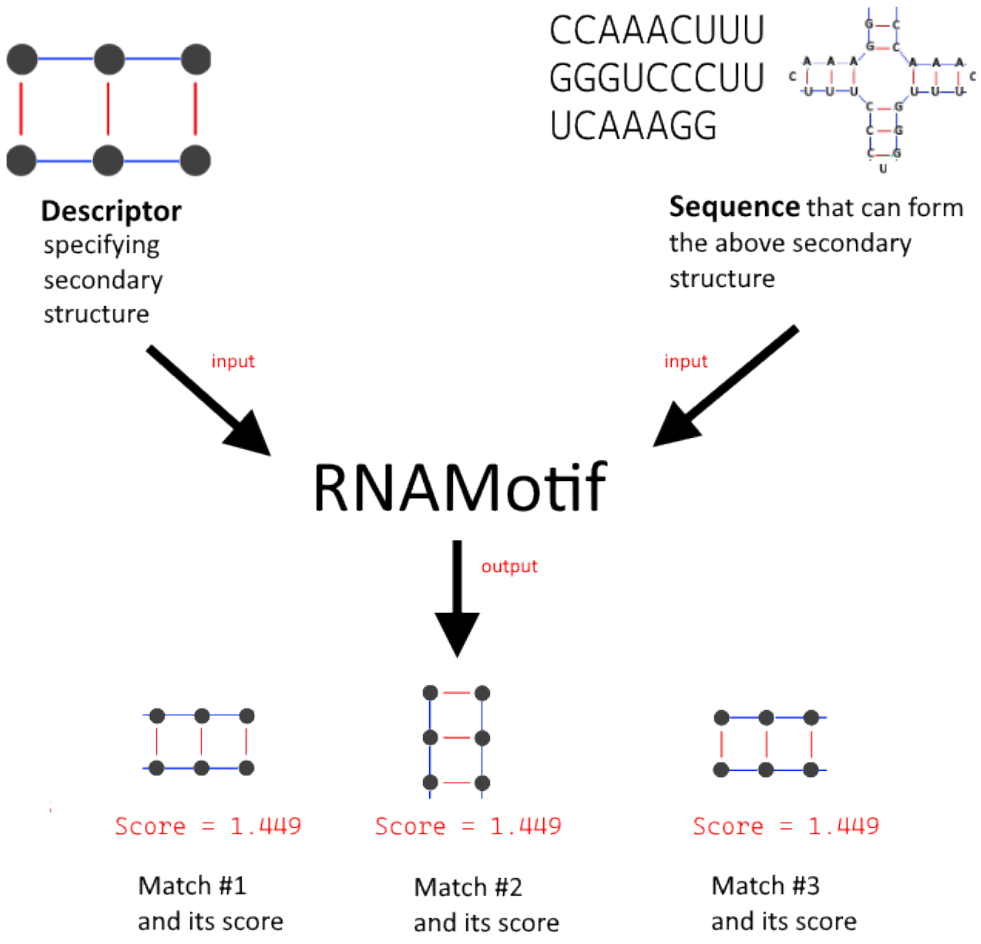

Figure 3.1: RNAMotif process: From a secondary structure descriptor and a sequence, to secondary structure matches.

```
parms
    chk_both_strs = 0;

descr
    h5 (tag = '0', minlen = 3, maxlen = 3)
        ss (minlen = 4, maxlen = 4)
    h3 (tag = '0')

score {
    SCORE = bits (h5['0'] ,h3['0']);
}
```

Figure 3.2: An example of a RNAMotif descriptor describing a hairpin loop. The first part ("parms") defines options, the second part ("descr") defines the secondary structure, while the third part ("score") defines how we want matches to be scored (using the "bits" function in this example).

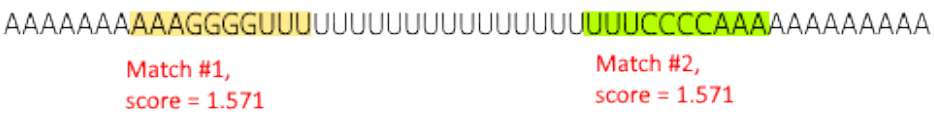

Figure 3.3: Example matches (highlighted) produced by RNAMotif using the descriptor in Figure 3.2 given an input sequence.

The RAG representations are undirected [50], meanwhile, the descriptors for RNAMotif take the direction of the RNA strand in the secondary structure into account [96]. This means, for certain graph representations, there will be more than one corresponding descriptors if different traversals of the same graph produced a different specific secondary structure. This difference is illustrated in Figure 3.4 and Figure 3.5. The first traversal of the a graph representation consisting of 3 vertices in Figure 3.4 would produce a secondary structure specification that starts with a helix, followed by bulge, another helix, hairpin loop, and so on. Meanwhile, the second and different traversal of the same graph shown in Figure 3.5 would result in a secondary structure specification starting with a helix, then a hairpin loop, another helix, a bulge, and so on. In short, a single RAG representation may have more than one different RNAMotif descriptors when the same graph is traversed differently.

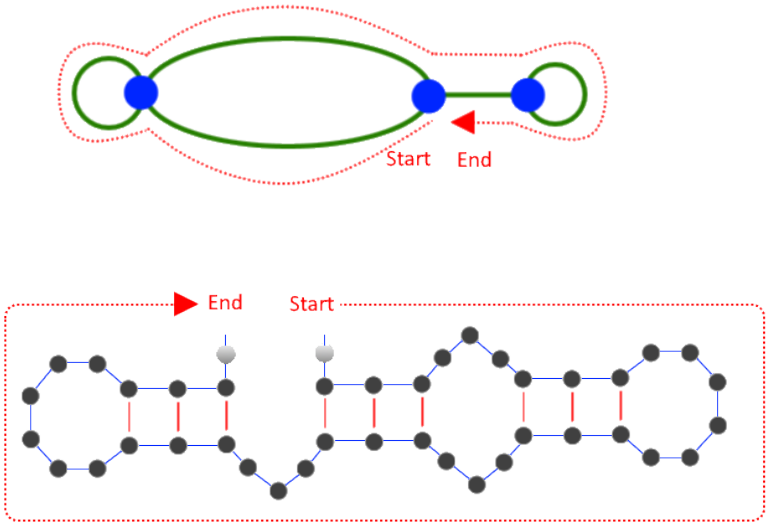

Figure 3.4: A sample traversal of a RAG representation (top) and its secondary structure (bottom).

After the descriptors are produced for dual graph representations with 2 to 5 vertices, structure matches and their scores can be obtained from a sequence using the program. When it comes to scoring matches, we used the "bits" scoring function built into RNAMotif, which supports pseudoknots and scores the matches based on the complexity of the sequence that contributes to the formation of the secondary structure [96, 134, 153].

The matches and their scores are the building blocks of the secondary structure fingerprints

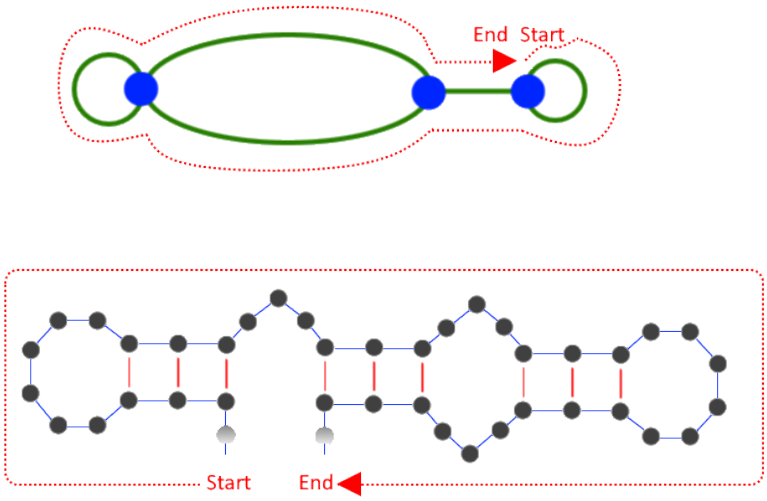

Figure 3.5: Another sample traversal of the same RAG representation (top) and its secondary structure (bottom).

as follows [134]:

1. Secondary structure matches and their scores of a sequence for each graph (i.e. graph #1 to graph #47) are obtained using RNAMotif and the corresponding descriptors;
2. For each RNAMotif descriptor (representing secondary structure from a specific traversal of one of the graphs), if there are more than 1 match resulting from the sequence, the match scores are averaged;
3. If there were no match against a secondary structure descriptor, the match score for that descriptor and the sequence is set to 0;
4. For graph representations with more than 1 corresponding descriptor, the scores are averaged as well to produce overall match score of that graph;
5. At this point, for a single sequence, we have 47 scores (i.e. score #1 to score #47 corresponding to graph #1 to graph #47 respectively);
6. These scores are then normalized so that they range between 0 and 1 inclusively (i.e. minimum score is 0, maximum score is 1, and scores in between are scaled accordingly in

a linear fashion).

7. The normalized values **become the secondary structure fingerprints** to be used as deep learning input features.

The steps to produce the averaged scores for each RAG graph given a sequence are illustrated in Figure 3.6.

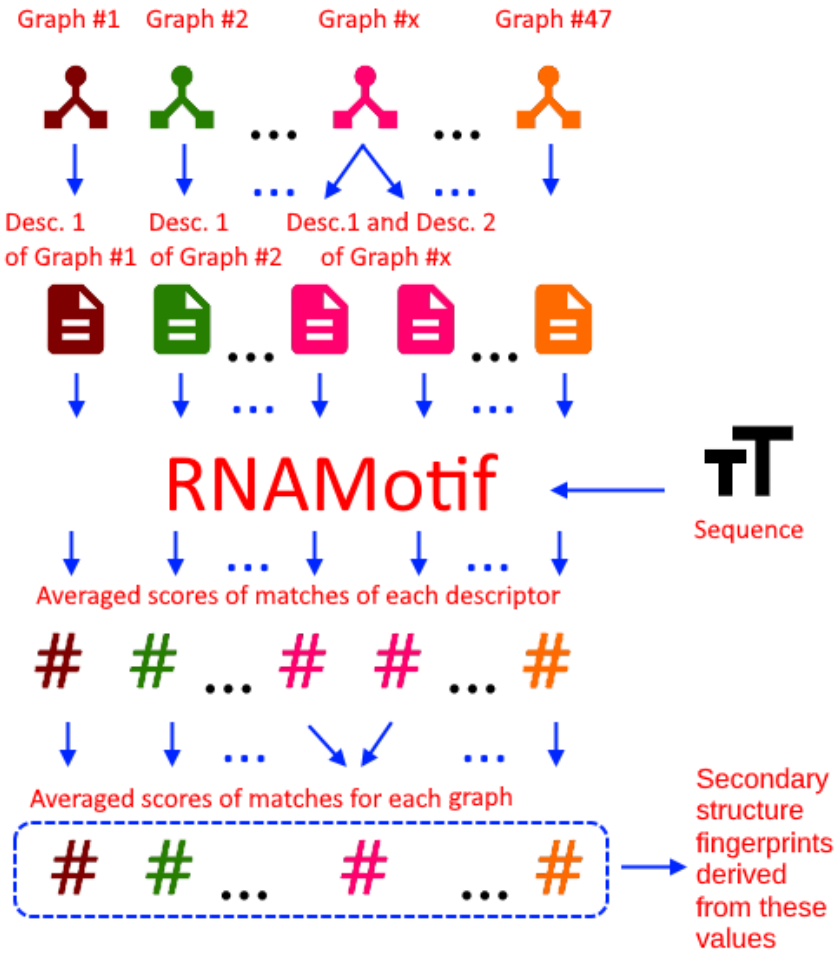

Figure 3.6: From RAG representations to secondary structure fingerprints. The colours on the graph, descriptor, and score icons represent which graph do the descriptors and scores belong to.

3.2 Common Secondary Structure Motif Fingerprints

The RAG-based secondary structure fingerprints do not contain nor represent information about the thermodynamic stability or free energy of the matching secondary structure motifs, as no free energy calculation was involved. RNAMotif does not support free energy calculation when pseudoknots are involved in the secondary structures, which is the case for some of the dual graphs representations [50, 96]. Therefore, in order to incorporate free energy values (which may provide additional information about the RNA function [32, 71]) into the fingerprints, we developed a separate set of secondary structure fingerprints based on common, known RNA secondary structure motifs/patterns [134]. In addition to providing information on thermodynamic stability, this additional set of fingerprints focuses on smaller/local secondary structure matches. Some of the curated motifs also take sequence information into account.

The first step to build the fingerprints in this set is curating the common known RNA secondary structure motifs to be used [134]. This was done by consulting the Rfam database [76, 77], and various previous studies on RNA secondary structure motifs [54, 69, 79, 83, 87, 89, 90, 104, 110, 131, 138, 139, 142, 159].

The second step is splitting the collected known secondary structure motifs into different groups. This is done so that the simpler motifs with only basic RNA secondary structure elements (such as a hairpin loop) will not be overly generic relative to the other motifs. In other words, there will not be overly basic motifs that would otherwise have excessively high chances of producing matches (i.e. almost always or always matches – thus are not as useful to make a distinction between different sequences). This was done by analyzing secondary structure frequency in the bpRNA database [37], which contains annotated RNA secondary structure data. This involves parsing the database in order to extract distributions of different lengths and sides/locations (i.e. 5' or 3' side) of the common RNA secondary structure motifs. The extracted distributions are illustrated in Figure 3.7, Figure 3.8, Figure 3.9, Figure 3.10, Figure 3.11, Figure 3.12, and Figure 3.13.

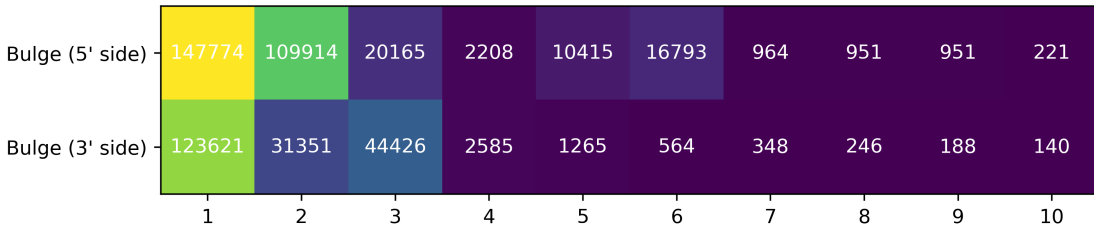

Figure 3.7: Produced heatmap from parsing the bpRNA database [37] illustrating the distribution of different bulge lengths on each side. The number in each cell is the number of occurrences of the corresponding bulge in the database.

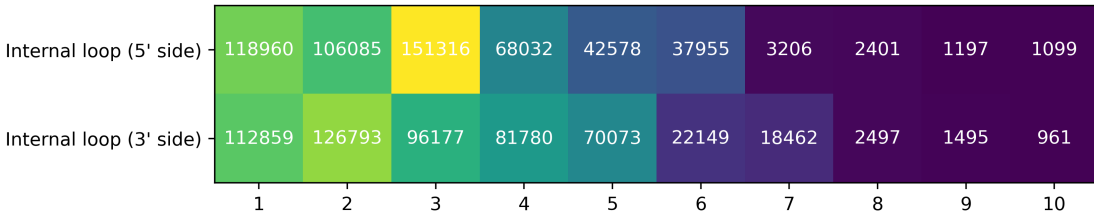

Figure 3.8: Produced heatmap from parsing the bpRNA database [37] illustrating the distribution of different internal loop lengths on each side. The number in each cell is the number of occurrences of the corresponding internal loop in the database.

Figure 3.9: Produced heatmap from parsing the bpRNA database [37] illustrating the distribution of different internal loop length combinations considering both sides at the same time (i.e. distribution of length pairs consisting of length on the 5' side and length on the 3' side. The number in each cell is the number of occurrences of the corresponding internal loop in the database.

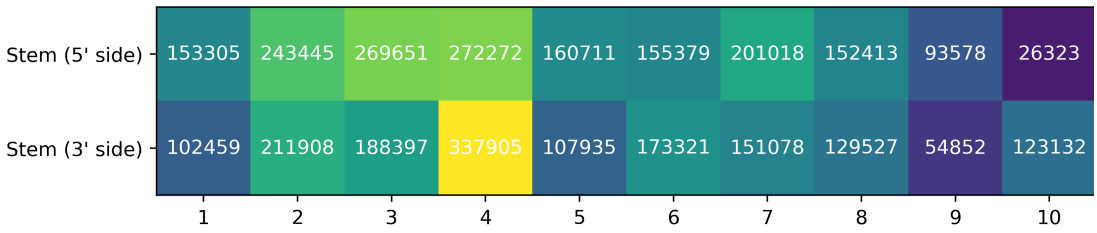

Figure 3.10: Produced heatmap from parsing the bpRNA database [37] illustrating the distribution of continuous paired nucleotides lengths of stems on each side. The number in each cell is the number of occurrences of the corresponding stems in the database.

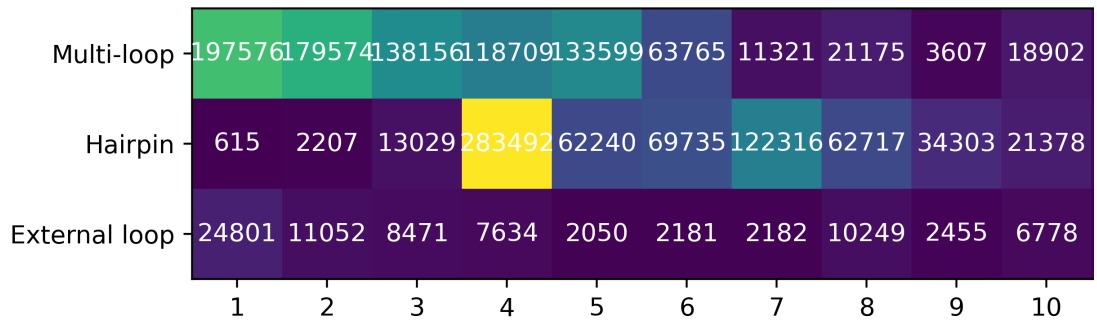

Figure 3.11: Produced heatmap from parsing the bpRNA database [37] illustrating the distribution of different lengths of different types of loops. The number in each cell is the number of occurrences of the corresponding loop in the database.

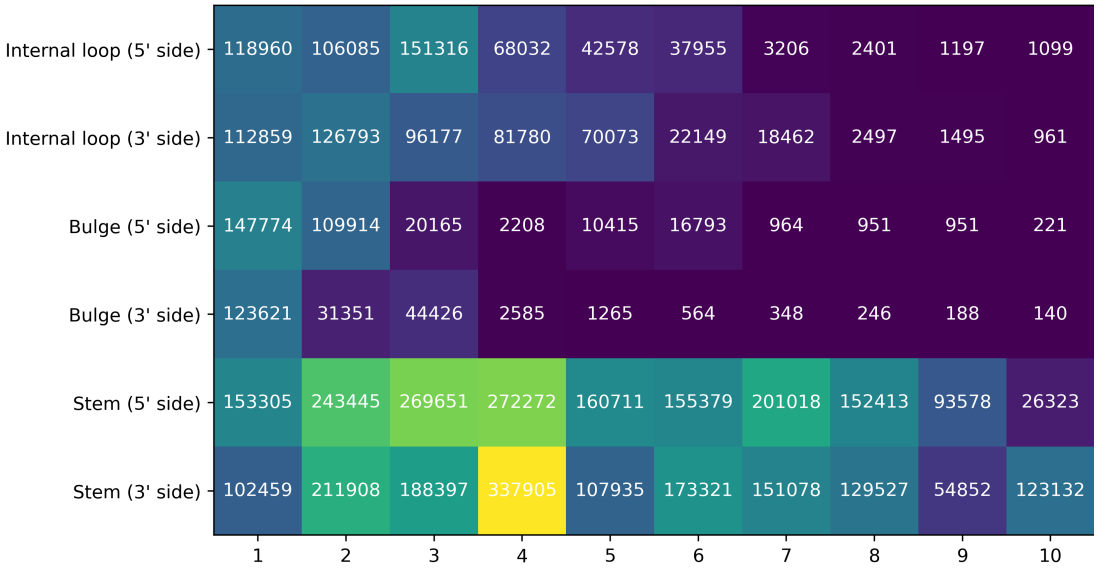

Figure 3.12: Produced heatmap from parsing the bpRNA database [37] illustrating the frequency distribution of different secondary structure elements with their corresponding lengths on each side. The number in each cell is the number of occurrences of the corresponding structure in the database.

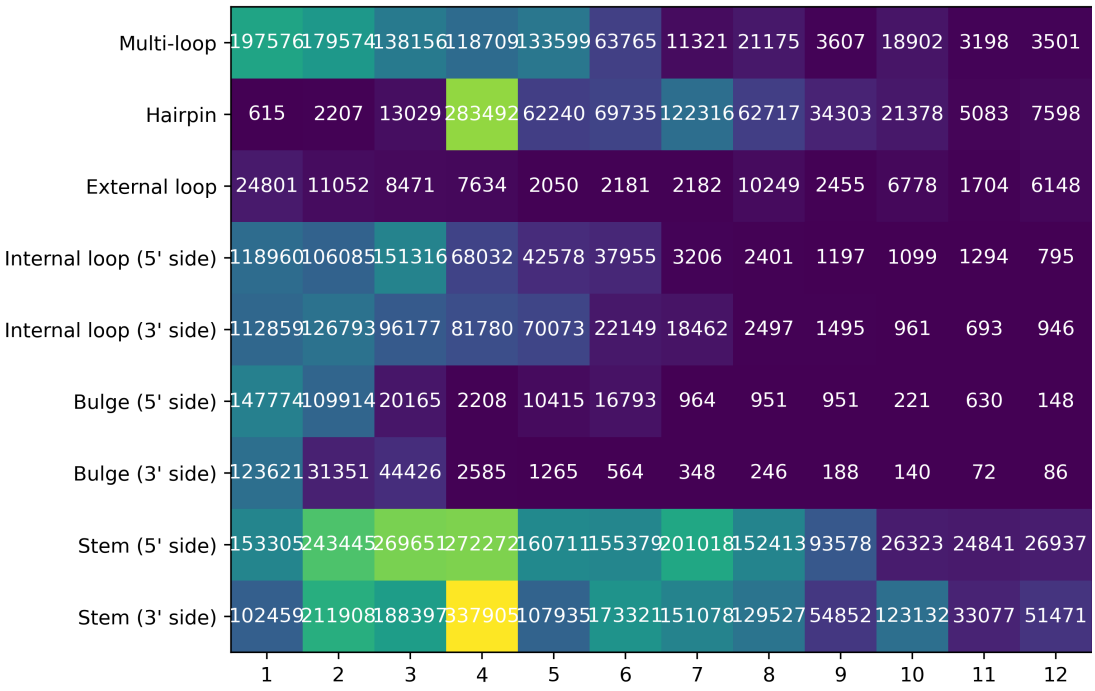

Figure 3.13: Produced heatmap from parsing the bpRNA database [37] illustrating the frequency distribution of all secondary structure elements and their corresponding lengths, relative to each other. The number in each cell is the number of occurrences of the corresponding structure in the database.

With these distributions, we then split the simpler secondary structure motifs into different length ranges and sides. For example, instead of including a bulge of any length and sides which would produce excessive matches; the bulge are broken down into 12 different specific motifs based on their lengths, sides (i.e. 5' or 3'), and in 2 cases, sequence.

At the end of the previously discussed steps and processes, our curated structural motifs used to build the common motifs based secondary structure fingerprints are as follows [134]:

1. "Hairpin with a loop length of 3 nucleotides" [134],
2. "Hairpin with a loop length of 4 nucleotides whose sequence is GNRA (the 'GNRA tetraloop' [69, 90])" [134],
3. "Hairpin with a loop length of 4 nucleotides whose sequence is UNCG (the 'UNCG tetraloop' [90, 104])" [134],
4. "Hairpin with a loop length of 4 nucleotides whose sequence is UMAC (the 'UMAC tetraloop' [159])" [134],
5. "Hairpin with a loop length of 4 nucleotides whose sequence is CUUG" [134, 139, 142],
6. "Hairpin with a loop length of 4 nucleotides whose sequence is CUCG" [134, 139],
7. "Hairpin with a loop length of 4 nucleotides whose sequence is GANC (the 'GANC tetraloop' [79])" [134],
8. "Hairpin with a loop length of 4 nucleotides whose sequence is ANYA" [83, 110, 134],
9. "Hairpin with a loop length of 4 nucleotides with any sequence" [134],
10. "Hairpin with a loop length of 5 nucleotides" [134],
11. "Hairpin with a loop length of 6 nucleotides" [134],
12. "Hairpin with a loop length of 7 nucleotides" [134],
13. "Hairpin with a loop length of 8 nucleotides" [134],
14. "Hairpin with a loop length of 9 nucleotides" [134],
15. "Hairpin with a loop length of 10 nucleotides" [134],
16. "Internal loop with length of 1 to 7 nucleotides on either side (5' or 3')" [134],
17. "Internal loop with a length of 1 nucleotide on the 5' side and a length of 1 nucleotide on the 3' side" [134],

18. “Internal loop with a length of 2 nucleotides on the 5’ side and a length of 1 nucleotide on the 3’ side” [134],
19. “Internal loop with a length of 1 nucleotide on the 5’ side and a length of 2 nucleotides on the 3’ side” [134],
20. “Internal loop with a length of 2 nucleotides on the 5’ side and a length of 2 nucleotides on the 3’ side” [134],
21. “Internal loop with a length of 3 nucleotides on the 5’ side and a length of 3 nucleotides on the 3’ side” [134],
22. “Internal loop with a length of 3 nucleotides on the 5’ side and a length of 4 nucleotides on the 4’ side” [134],
23. “‘C-loop’ on the 5’ side” [89, 134],
24. “‘C-loop’ on the 3’ side” [89, 134],
25. “Internal loop with ‘tandem AG pairs’ [54]” [134],
26. “Internal loop with the ‘UAA/GAN motif’ [87]” [134],
27. “Bulge with a length of 1 nucleotide on the 5’ side” [134],
28. “Bulge with a length of 2 nucleotides on the 5’ side” [134],
29. “Bulge with a length of 3 nucleotides on the 5’ side” [134],
30. “Bulge with a length of 4 nucleotides on the 5’ side” [134],
31. “Bulge with a length of 5 nucleotides on the 5’ side” [134],
32. “Bulge with a length of 6 nucleotides on the 5’ side” [134],
33. “Bulge with a length of 1 nucleotide on the 3’ side” [134],
34. “Bulge with a length of 2 nucleotides on the 3’ side” [134],
35. “Bulge with a length of 3 nucleotides on the 3’ side” [134],
36. “Bulge with a length of 4 nucleotides on the 3’ side” [134],
37. “Bulge with the ‘Sarcin/Ricin’ motif [131, 138] on the 5’ side” [134],
38. “Bulge with the ‘Sarcin/Ricin’ motif [131, 138] on the 3’ side” [134],
39. “Paired stem with a length of 3 to 4 nucleotides” [134],

40. "Paired stem with a length of 5 to 6 nucleotides" [134],
41. "Paired stem with a length of 7 to 8 nucleotides" [134],
42. "Paired stem with a length of 9 to 10 nucleotides" [134],
43. "Paired stem with a length of 11 nucleotides" [134],
44. "Paired stem with a length of 12 nucleotides" [134].

Similar to our approach with the RAG-based secondary structure fingerprints, RNAMotif [96] was used to find matches of each curated motif and score the matches. Thus, the descriptor for each of the motifs were made. And unlike the case with the RAG-based descriptors where each graph representation may have more than one descriptor, each of the curated secondary structure motifs only corresponds to a single RNAMotif descriptor. In addition, when it comes to scoring the matches, free energy calculation is used instead of the "bits" scoring system used with the RAG-based descriptors [96, 134] – in particular, the "efn2" function in RNAMotif, which is an implementation of a free energy calculation approach devised and proposed by Mathews, et al. [98], was used for this purpose [96, 134].

Thus, using the curated motifs, the steps to produce the secondary structure fingerprints from RNA sequences are as follows:

1. For each sequence in the dataset, secondary structure matches and their free energy values for each of the curated motifs are obtained using RNAMotif [96];
2. For each motif with more than 1 match resulting from the sequence (i.e. the motif can be found at different positions), the minimum, average, and maximum free energy values are kept (3 separate values). Otherwise, if there is only 1 match for that motif, the minimum, average, and maximum free energy values are set to the free energy value of that 1 match;
3. After processing all of the sequences in the dataset, non-matches for each motif are given the highest (representing poorest or least thermodynamically stable match) minimum, average, and maximum free energy values the motif (not the individual sequence) has resulted in across all of the sequences in the dataset (i.e. using and keeping the 3 different values separately);
4. At this point, there are 3 values for each sequence and each motif, which are then separately rescaled.
5. For each of the motif across all of the sequences in the dataset: lowest free energy (best match) for each of the minimum, average, and maximum values are set to 1, the highest (poorest match) for each of the minimum, average, and maximum values are set to 0, and

the rest of the values are rescaled accordingly, again, separately for the minimum, average, and maximum values (i.e. the rescaling for the minimum values does not take into account any values in the average or maximum values, and likewise with the average and maximum values);

6. We then have 3 sets of rescaled values (minimum, average, maximum free energy matches), each of the set containing 44 values per sequence representing the 44 different motifs; which can then be used separately or combined as the deep learning input features.

We initially only used the minimum free energy when there are multiple matches (hence only 1 set of values consisting of the 44 values per sequence) [134]; but found out in a later study that using and including the average and maximum in addition to the minimum free energy, when there are more than 1 matches in a sequence for a descriptor, resulted in an improved classification performance [135]. In addition, in one of the studies [134], two different base pairing rules were used in order to produce two separate set of fingerprints based on the common motifs: one that only allows Watson-Crick base pairs without allowing G·U wobble pairs, and another that allows both Watson-Crick base and G·U wobble pairs [134, 146].

3.3 K-mers

The k-mer representations used in our studies contains information about subsequences (of length k) distributions, in which the subsequences are made of the following 4 nucleotide letters: "A" for adenine, "C" for cytosine, "G" for guanine, and either "T" or "U" for uracil. The process of generating k-mer representations of a dataset begins with producing an ordered list containing possible subsequences of length k with different combinations of the possible 4 nucleotide letters. This list and its order of subsequences are used with all sequences in the dataset, such that each position in all of the resulting representations would correspond to the same subsequence, which is important when building features for machine/deep learning. For example, for $k = 3$, the first position of all k-mer representations derived from different sequences in the dataset always corresponds to "AAA", while the second to "AAC", and so on. Using this ordered list, the following outlines the remaining steps to produce a k-mer representation of a sequence in the dataset:

1. The occurrences of each possible subsequence of length k in the aforementioned list are counted from the sequence;

2. Each of the occurrence counts are then divided by the total number of occurrences of all possible subsequences, such that we now have a list of values representing the distribution of the different possible subsequences that add up to 1. This list of values are then used as the deep learning input vector representing the sequence.

We used $k = 2, 3, 4$ with one study [134], and $k = 4, 5, 6$ in another [135].

3.4 Skip-mers

"Skip-mers" [31] were used in one of our studies in order to capture longer but non-exact sequence patterns that may exist in the dataset [135]. Similar to the k-mer representations in our studies, 4 possible nucleotides were used to build the possible combinations of skip-mers: "A", "C", "G" and "U"/"T".

Two sets of match-skip rules were used: a set uses 1 matching character followed by 1 skipped character, and another set uses 2 consecutive matching characters followed by 1 skipped character [135]. The latter set is intended to capture any patterns that may exist in "the first two nucleotides of codons", as the third nucleotide often does not affect the resulting amino acid (i.e. may be redundant) and therefore skipped [56,135]. If such patterns indeed exist and are efficiently captured in the skip-mer representations, without being masked by noises that may result from "false matches from non-coding regions" [135] and/or false matches resulting from wrong "open reading frames" [105,135], they may serve as additional useful deep learning features that could result in improvements to the classification performance [135].

In our study that utilizes skip-mers, we used the same number of matching characters as the k-mer representations in that study: i.e. $matching\ characters = 4, 5, 6$ as $k = 4, 5, 6$. Thus, given these numbers of matching characters and the 2 sets of match-skip rules, the skip-mers we used in the study are as follows:

- "Match 1 Skip 1 with a length of 7 (4 matching characters in total)" [135],
- "Match 1 Skip 1 with a length of 9 (5 matching characters in total)" [135],
- "Match 1 Skip 1 with a length of 11 (6 matching characters in total)" [135],
- "Match 2 Skip 1 with a length of 6 (4 matching characters in total)" [135],
- "Match 2 Skip 1 with a length of 7 (5 matching characters in total)" [135], and
- "Match 2 Skip 1 with a length of 9 (6 matching characters in total)" [135].

The steps to produce a specific skip-mer representation for a specific sequence is similar to the steps involved to produce a k-mer representation. First, an ordered list containing combinations of possible matching characters is generated. Then, for a sequence, the number of occurrences of each combination of matching characters (at their corresponding positions according to the skip-mer configuration) is counted. Afterwards, each of the counts is divided by the total of

these counts, producing a distribution of matches that adds up to 1 when summed. These values representing the distribution of occurrences then become the input features representing the RNA sequence. Finally, just like the k-mer representations, the order of the values are kept consistent such that the first value corresponds to the first combination of matching characters in the list, the second value corresponds to the second combination, and so on.

3.5 Deep Learning

All of our deep learning implementation was done with Keras [30], utilizing the Tensorflow [1] backend [134, 135]. Deep neural networks with different specific configurations were used in the different studies [134, 135], as the datasets and the problems to solve differed as well. In other words, each of the different deep neural network configurations, including hyperparameters such as the number of epochs, were optimized for and therefore is specific to each of the datasets. Thus, details about the specific configurations and hyperparameters are provided in the corresponding dataset sections – Section 4.1.3 and Section 4.2.4.

The optimal specific configurations were determined by performing manual experimentation on various configurations. For example, deeper network configurations compared to the ones in our studies were not used; as we determined that either they failed to provide additional increase in classification performance, or introduced overfitting.

In addition, while experimenting with the different possible configurations for each of the different datasets, we plotted and monitored the accuracy and loss graphs of the training and test sets across the different epochs, in order to ensure that overfitting is minimized with the chosen model configurations. The training set consists of 90% of the data, while the test set has the remaining 10%. Example graphs for the model used in our non-coding RNA classification study [134] are shown in Figure 3.14, Figure 3.15, Figure 3.16, Figure 3.17, and Figure 3.18 for the RAG-based fingerprints, common motifs based fingerprints without wobble, common motifs based fingerprints with wobble, all types of secondary structure fingerprints combined, and all secondary structure fingerprints with 4-mer combined respectively. Specifically, we employed regularizations [33] in the chosen deep learning model of this study to minimize overfitting [134], as further discussed in Section 4.1.3.

Similarly, in our RNA virus study [135], we also plotted the graphs to check for overfitting. Example graphs from this study are shown in Figure 3.19 (using 4-mer), Figure 3.20 (using match-1-skip-1 skip-mer with a length of 9), and Figure 3.21 (using common motifs based secondary structure fingerprints derived from the minimum, average, and maximum match scores).

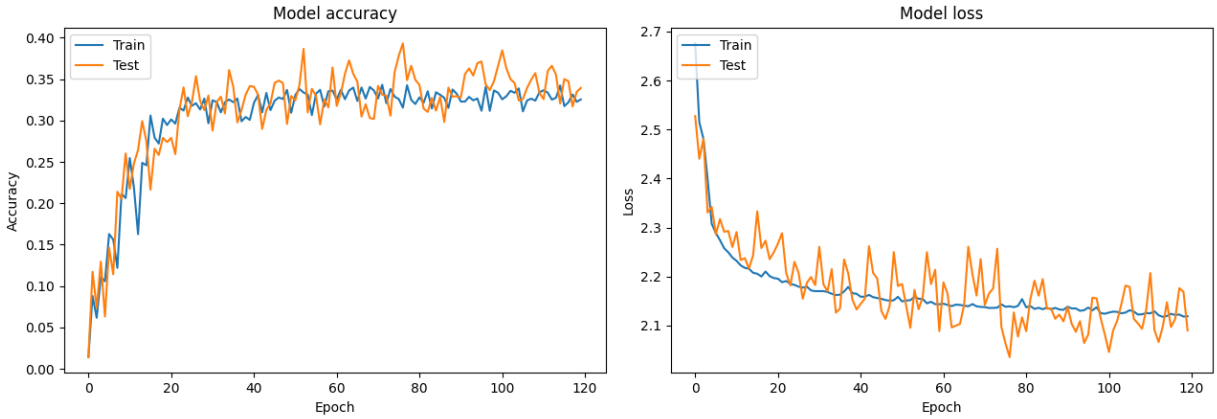

Figure 3.14: Accuracy and loss plots of the RAG-based fingerprints throughout model training in the non-coding RNA study.

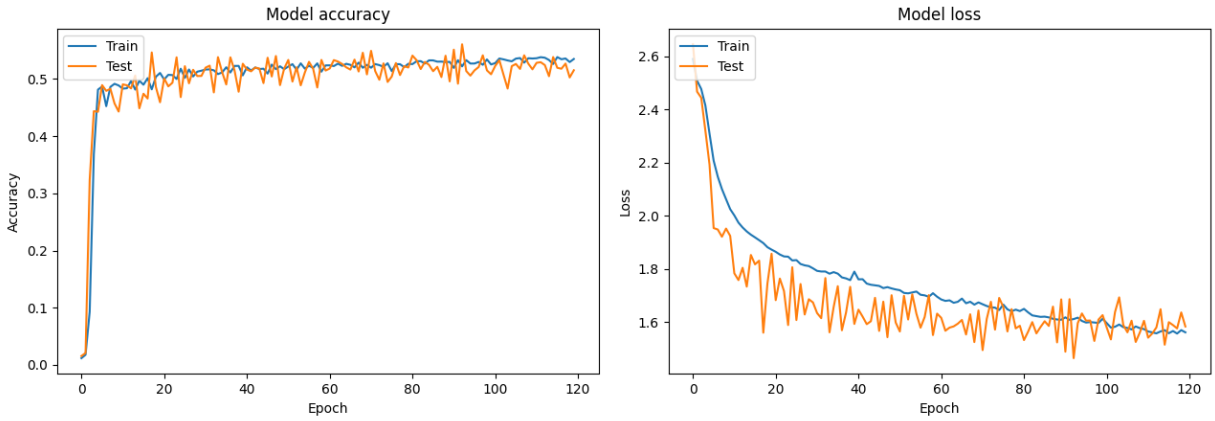

Figure 3.15: Accuracy and loss plots of the short motifs based fingerprints without wobble throughout model training in the non-coding RNA study.

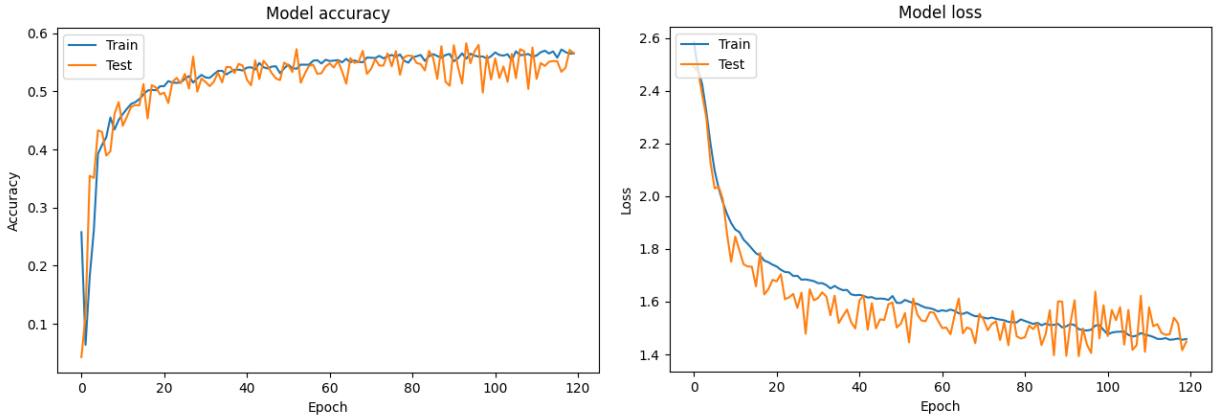

Figure 3.16: Accuracy and loss plots of the short motifs based fingerprints with wobble throughout model training in the non-coding RNA study.

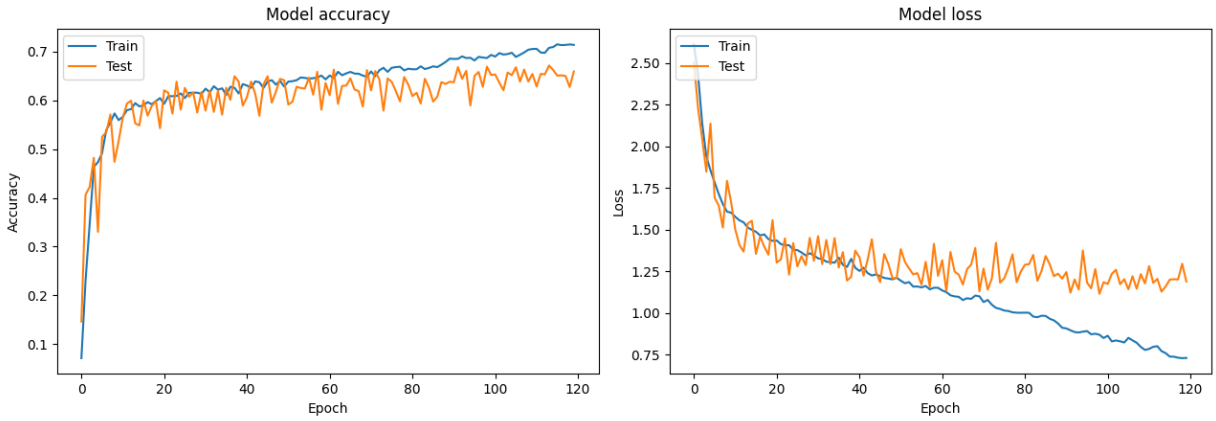

Figure 3.17: Accuracy and loss plots using all secondary structure fingerprints combined throughout model training in the non-coding RNA study.

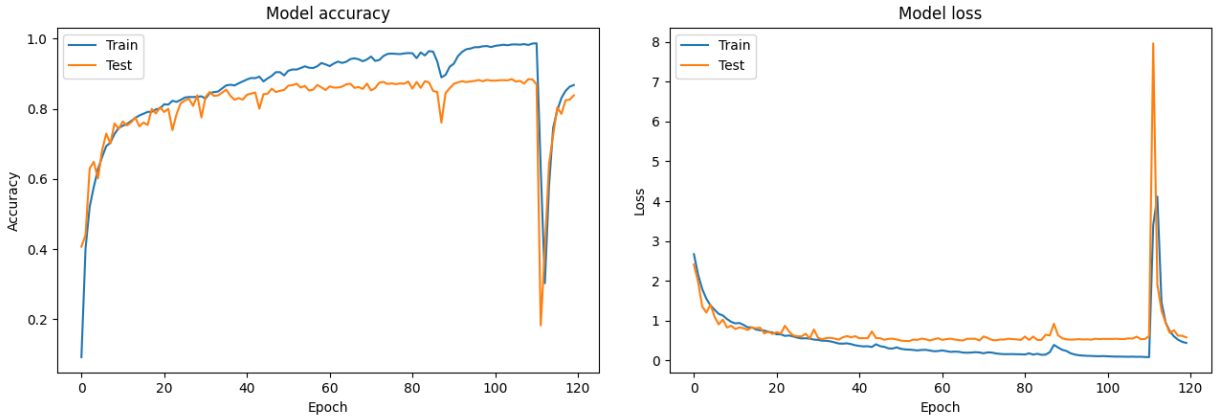

Figure 3.18: Accuracy and loss plots using all secondary structure fingerprints and 4-mer combined throughout model training in the non-coding RNA study.

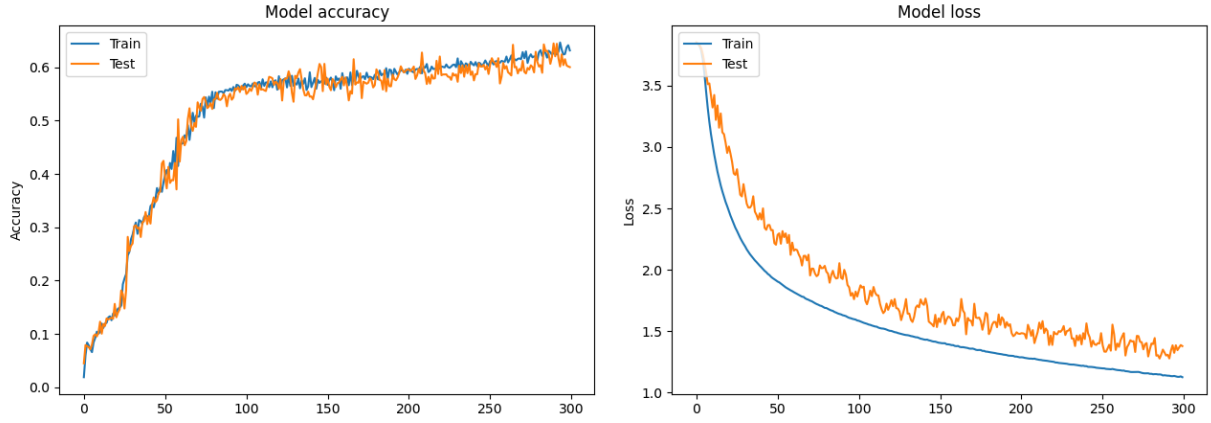

Figure 3.19: Accuracy and loss plots using 4-mer combined throughout model training in the RNA virus study.

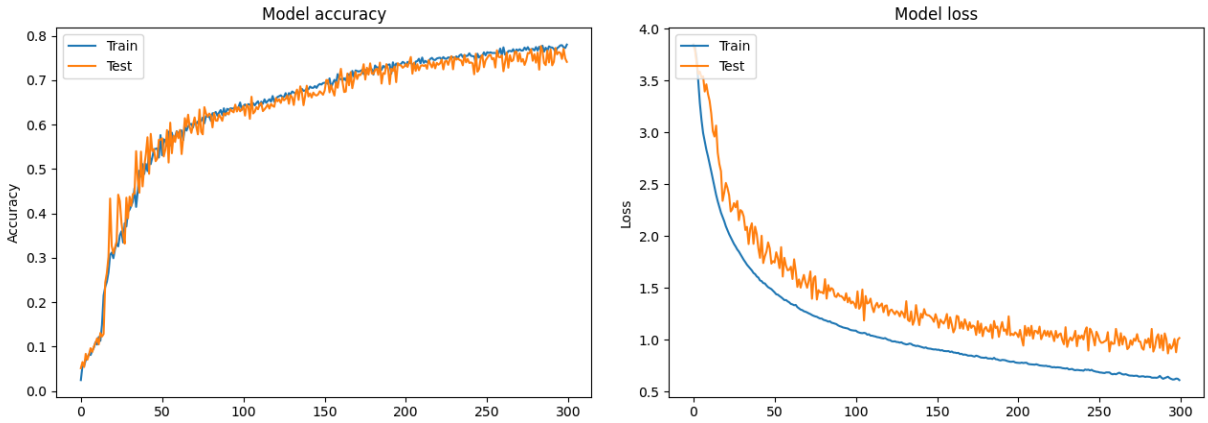

Figure 3.20: Accuracy and loss plots using match-1-skip-1 skip-mer of length 9 throughout model training in the RNA virus study.

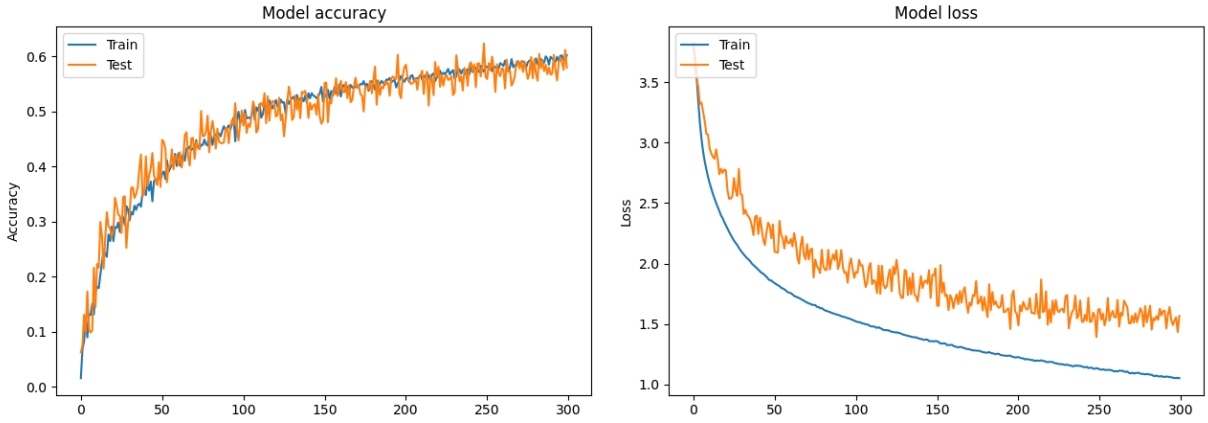

Figure 3.21: Accuracy and loss plots using common motifs based secondary structure fingerprints (using minimum, average, and maximum match scores combined) throughout model training in the RNA virus study.

All of the deep neural networks in both studies utilize dense connections between the different layers [134, 135]. Alternative configurations, such as "convolutional neural network" (CNN) [4] or "long short-term memory" (LSTM) [60], do not apply to our input features. This is because the features are not meant to be down-sampled (thus do not suit CNN), and are not sequential (thus are not suitable with LSTM).

Chapter 4

Datasets

4.1 Non-Coding RNAs from Rfam 14.1

4.1.1 Overview

Rfam provides a database containing RNA sequences and their classes/types [76, 77], which was used by one of our studies to assess the effectiveness of secondary structure fingerprints and sequence-based features with deep learning. The dataset was based on Rfam version 14.1, but several filtering steps were involved so that the dataset used to evaluate the classification performance is non-redundant [76, 77, 134].

First, we removed duplicate entries from the raw, downloaded dataset [76, 77, 134]. We also found out that longer sequences require large computational resources for RNAMotif to process given the RAG-based descriptors, especially memory requirements. Thus, longer sequences that could not be processed by RNAMotif with a reasonable amount of computational resources were not included – this limitation is further discussed in Section 6.1. At this point, we have 1,013,200 entries of RNA sequences, their classes, and their secondary structure fingerprints [134]. This is about 51.78% compared to the number of unique sequences in the Rfam 14.1 dataset, which is 1,956,735.

Next, in order to reduce redundancy in the dataset, the CD-hit tool was used [48, 91] to exclude sequences that are more than 80% similar to one another from the dataset [134]. This CD-hit [48,91] threshold of 80% was also used by other previous studies involving RNA sequences [111, 134, 154]. After this step, we ended up with 63,612 sequences [134].

Finally, while the Rfam 14.1 dataset [76, 77] contains 31 non-coding RNA classes, not all of the classes are included in the study. Particularly, classes with too few or no sequences after

the previous filtering steps were excluded, yielding 59,723 sequences and 12 classes. Each of the sequence belongs to one of the following classes: "Cis-reg, miRNA, CD-box, ribozyme, snRNA, HACA-box, Intron Group II, tRNA, 5S rRNA, sRNA, antisense, and riboswitch" [134]. Similar to a related study that uses only 13 of the Rfam non-coding RNA classes [46], but unlike approaches that use the excluded sequences and combine them to a single class (e.g. "other" class), our current study did not attempt to use or analyze the excluded sequences and classes.

4.1.2 Deep Learning Features

The following set of features were used and evaluated in the study [134]:

- RAG-based secondary structure fingerprints;
- Fingerprints based on the curated secondary structure motifs without allowing wobble pairs;
- Fingerprints based on the curated secondary structure motifs allowing wobble pairs;
- Combination (used together) of the common secondary structure motifs fingerprints that allow and disallow wobble pairs;
- Combination of the RAG-based fingerprints, common secondary structure motifs fingerprints that allow wobble pairs, and common secondary structure motifs fingerprints that disallow wobble pairs;
- 2-mer;
- 3-mer;
- 4-mer;
- Combination of the RAG-based fingerprints, common secondary structure motifs fingerprints that allow wobble pairs, common secondary structure motifs fingerprints that disallow wobble pairs, and 2-mer;
- Combination of the RAG-based fingerprints, common secondary structure motifs fingerprints that allow wobble pairs, common secondary structure motifs fingerprints that disallow wobble pairs, and 3-mer; and
- Combination of the RAG-based fingerprints, common secondary structure motifs fingerprints that allow wobble pairs, common secondary structure motifs fingerprints that disallow wobble pairs, and 4-mer.

4.1.3 Deep Learning Configuration

The deep neural network used in this study consists of 5 layers, and each layer is "densely-connected" with the next layer [134]. These layers are as follows [134]:

1. A starting input layer with n neurons (i.e. with a width of n), where n is the size of the input feature representing the sequences (e.g. $n = 47$ for the RAG-based secondary structure fingerprints, $n = 44$ for the fingerprints based on the curated common secondary structure motifs, and $n = 47 + 44$ when former example is used together with the latter example);
2. A ReLU-activated [42] layer with $n \times 3$ neurons;
3. A ReLU-activated layer with the same number of neurons as the previous layer (i.e. $n \times 3$ neurons), with L2 regularization [33] of 0.0001 applied to the kernels;
4. Another ReLU-activated layer of width $n \times 3$, with the same L2 regularization of 0.0001;
5. A sigmoid-activated [109] layer consisting of $n \times 3$ neurons;
6. A final, softmax-activated [41] output layer with 12 neurons, where 12 is the number of possible classes that an input RNA from the dataset can be classified into.

These layers are illustrated in Figure 4.1.

The network configuration was chosen for this problem after performing assessments of various possible configurations; including deeper (more layers), more shallow (less layers), wider (larger number of neurons in a layer), and less wide (lesser number of neurons in a layer) networks; and different hyperparameters including activation functions [134].

4.1.4 Training Hyperparameters and Evaluation

For each of the feature sets, a corresponding deep neural network is trained from scratch for 120 epochs, starting with a learning rate of 0.001 that decays by a factor of 0.1 every 40 epochs (i.e. learning rate is 0.001 in epoch 1 to 40, 0.0001 in epoch 41 to 80, and 0.00001 in epoch 81 to 120) [134]. This decay is in place as our preliminary training without it (i.e. with a constant learning rate instead) attained a lower maximum classification accuracy. Indeed, according to a recent study on "learning rate decay", using such decay allows the models to better learn the more "complex patterns" [156].

In each of the training and evaluation session, 90% of the data is used to train the model, while the remaining 10% is used to evaluate the classification performance of the trained model.

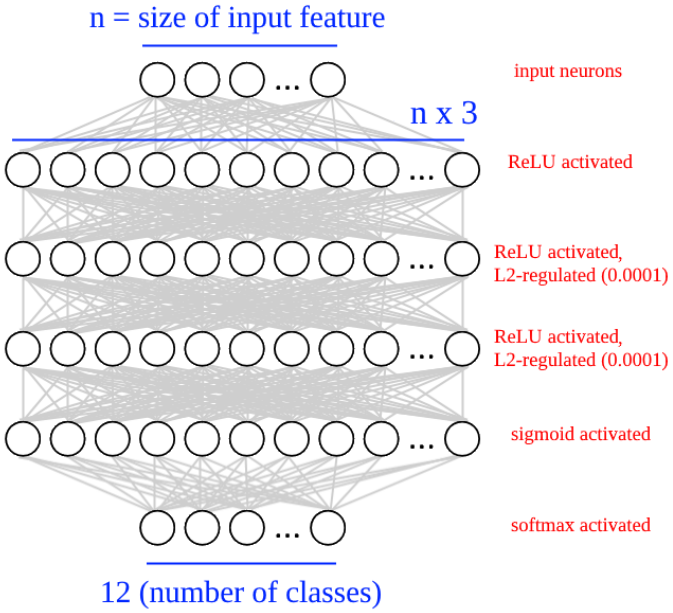

Figure 4.1: Deep learning architecture for the Rfam 14.1 [76, 77] dataset.

In addition, as previously discussed, since using a specific combination/split of training and evaluation data may result in non-representative results, we employed 10-fold cross validation [12] – so that the training and evaluation processes are performed 10 times independently, each with a different training set and evaluation set. The results from all 10 folds are then averaged to get the overall representative result, which is then considered as the performance of the feature set being used.

4.2 RNA Virus Sequences and Their Host Species from NCBI Virus

4.2.1 Overview

For our other study that involves secondary structure fingerprints based on the curated common RNA secondary structure motifs, k-mers, and skip-mers; we used a dataset obtained from NCBI Virus [18], containing viral RNA sequences and the host species they infect [135].

Both sequence-based features and secondary structure based features should be relevant to

this dataset; as previous work in virus-host relationships have indeed used sequence information [3, 49, 102, 135, 158], and secondary structures are known to be involved in the lifecycle of "single-stranded RNA viruses" [62, 66, 127, 135].

We downloaded the RNA virus dataset consisting only of viral sequences that are annotated as "complete" on NCBI Virus, on September 12, 2020 [18, 135]. The following filtering steps were then performed [135]:

1. Removal of duplicate entries,
2. Exclusion of entries which sequence length is greater than 40,000 nucleotides (to allow RNAMotif to perform the secondary structure matching and scoring of the entire filtered dataset, with reasonable amount of computational resources),
3. Exclusion of host species with less than 100 viral sequences – to ensure that the training process is not impacted by insufficient quantity of data in some of the hosts, per the approach of a prior study on the association between virus and the host genera [158].

The filtered dataset after these steps consists of 47,266 entries, with the following 47 different host species: "*Allium sativum*, *Anas carolinensis*, *Anas clypeata*, *Anas platyrhynchos*, *Anatidae*, *Apodemus agrarius*, *Aves*, *Bos taurus*, *Canis lupus familiaris*, *Capra hircus*, *Capsicum annuum*, *Columbidae*, *Corvus brachyrhynchos*, *Cricetulus griseus*, *Culex*, *Culex pipiens*, *Culex quinquefasciatus*, *Culicidae*, *Culiseta melanura*, *Cyanocitta cristata*, *Equus caballus*, *Felis catus*, *Gallus gallus*, *Glycine max*, *Homo sapiens*, *Macaca mulatta*, *Malus domestica*, *Meleagris gallopavo*, *Melogale*, *Mus musculus*, *Oryza sativa*, *Ovis aries*, *Procyon lotor*, *Prunus*, *Prunus avium*, *Prunus persica*, *Pyrus communis*, *Rattus norvegicus*, *Rosa sp.*, *Solanum lycopersicum*, *Solanum tuberosum*, *Sus scrofa*, *Sus scrofa domesticus*, *Triticum aestivum*, *Vitis vinifera*, *Vulpes vulpes*, *Zea mays*" [135]. The length of the longest included sequence is 31,473 [135].

4.2.2 Additional CD-hit Reduced Dataset

In addition to our primary dataset, in order to assess the classification performance when the data is limited, we have used two additional reduced datasets [135]:

- The first reduced dataset contains only sequences "that are at most 90% similar" [135], as produced using the CD-hit [48, 91] tool;
- The second reduced dataset only retains sequences "that are at most 80% similar" [135], again produced using the CD-hit [48, 91] tool.

Just like our other study, the lowest threshold of 80% was chosen as it has been used in prior work which datasets consist of RNA sequences [94, 112, 134, 135, 154].

Finally, the host species that are included need to have at least 50 (instead of 100) viral RNA sequences in the 80% dataset [135]. The 14 host species that met the criteria and therefore were used with both of the reduced datasets are listed as follows: "*Allium sativum*, *Arthropoda*, *Astacoidea*, *Bos taurus*, *Culicidae*, *Gallus gallus*, *Hexapoda*, *Homo sapiens*, *Octopus*, *Odonata*, *Sarcosphaera coronaria*, *Solanum lycopersicum*, *Sus scrofa*, *Vitis vinifera*" [135].

After filtering to use only these host species, we ended with 3,163 entries in the 80% dataset, and 5,781 entries in the 90% dataset [135].

4.2.3 Deep Learning Features

The sets of features that were used and compared in this study are shown in Table 4.1.

Table 4.1: Feature sets used for the virus-host classification problem [135].

Feature Set	K-mer	"Skip-mer" [31]			Common Structure Fingerprints
		Length	Match	Skip	
1	4-mer		-		-
2	5-mer		-		-
3	6-mer		-		-
4	-	6	2	1	-
5	-	7	1	1	-
6	-	7	2	1	-
7	-	9	1	1	-
8	-	9	2	1	-
9	-	11	1	1	-
10	-		-		min. free energy
11	-		-		min., avg. free energy
12	-		-		min., avg., max. free energy
13	4-mer	6	2	1	-
14	4-mer	7	1	1	-
15	5-mer	7	2	1	-
16	5-mer	9	1	1	-
17	6-mer	9	2	1	-
18	6-mer	11	1	1	-
19	4-mer		-		min. free energy
20	5-mer		-		min. free energy
21	6-mer		-		min. free energy
22	4-mer		-		min., avg. free energy
23	5-mer		-		min., avg. free energy
24	6-mer		-		min., avg. free energy
25	4-mer		-		min., avg., max. free energy
26	5-mer		-		min., avg., max. free energy
27	6-mer		-		min., avg., max. free energy
28	-	6	2	1	min. free energy
29	-	7	1	1	min. free energy
30	-	7	2	1	min. free energy
31	-	9	1	1	min. free energy
32	-	9	2	1	min. free energy
33	-	11	1	1	min. free energy
34	-	6	2	1	min., avg. free energy
35	-	7	1	1	min., avg. free energy
36	-	7	2	1	min., avg. free energy
37	-	9	1	1	min., avg. free energy
38	-	9	2	1	min., avg. free energy
39	-	11	1	1	min., avg. free energy
40	-	6	2	1	min., avg., max. free energy
41	-	7	1	1	min., avg., max. free energy
42	-	7	2	1	min., avg., max. free energy
43	-	9	1	1	min., avg., max. free energy
44	-	9	2	1	min., avg., max. free energy
45	-	11	1	1	min., avg., max. free energy
46	4-mer	6	2	1	min. free energy
47	4-mer	7	1	1	min. free energy
48	5-mer	7	2	1	min. free energy
49	5-mer	9	1	1	min. free energy
50	6-mer	9	2	1	min. free energy
51	6-mer	11	1	1	min. free energy
52	4-mer	6	2	1	min., avg. free energy
53	4-mer	7	1	1	min., avg. free energy
54	5-mer	7	2	1	min., avg. free energy
55	5-mer	9	1	1	min., avg. free energy
56	6-mer	9	2	1	min., avg. free energy
57	6-mer	11	1	1	min., avg. free energy
58	4-mer	6	2	1	min., avg., max. free energy
59	4-mer	7	1	1	min., avg., max. free energy
60	5-mer	7	2	1	min., avg., max. free energy
61	5-mer	9	1	1	min., avg., max. free energy
62	6-mer	9	2	1	min., avg., max. free energy
63	6-mer	11	1	1	min., avg., max. free energy

4.2.4 Deep Learning Configuration

Unlike our other study, we found that certain sets of features produced better results when used with more shallow neural networks, whereas other sets performed better with deeper neural networks [135]. Therefore, for each of the feature sets, we decided to use 3 different neural network configurations that differ only in the total number of layers (i.e. the depth) – the best classification performance of a feature set resulting from one of the 3 different neural network depths is then considered as the performance of that feature set [135].

The different numbers of layers are 3, 4, and 5. These layers start with 2, 3, or 4 consecutive ReLU-activated [107] densely-connected layers (applies to the neural network configuration with a total depth of 3, 4, and 5 respectively), in which the number of neurons in each layer is equal to the total number of values carried by all of the features being used (e.g. the number of neurons of each layer would be 4^k if k-mer is being used on its own, or $44 + 4^k$ if k-mer and the secondary structure fingerprints derived from minimum free energy values are being used together, or $44+44$ if the secondary structure fingerprints derived from both minimum and average free energy values are used); followed by one last softmax-activated [41] layer, in which the number of neurons is equal to the number of possible host species that a viral sequence could be classified into (i.e. 47 when the primary dataset is used, 14 when either of the reduced datasets is used).

4.2.5 Training Hyperparameters and Evaluation

The training and evaluation with the primary non-reduced dataset and the reduced datasets were done differently.

With the primary dataset, we performed a 10-fold cross validation [12] for each of the feature sets and each of 3 neural network configurations with different depths [135]. In each fold of a specific feature set and a specific deep neural network configuration, a corresponding neural network model was trained for 300 epochs from scratch; using the default learning rate (0.001) for the first 100 epochs, half of the default learning rate (0.0005) for the next 100 epochs, and a quarter of the default learning rate (0.00025) for the last 100 epochs – i.e. the learning rate decays by a factor of 0.5 every 100 epochs [135]. In addition, in each fold, 90% of the data was used for training, while the remaining 10% was used to evaluate the performance of the model and the feature set post-training [135]. Since 10-fold cross validation was performed, the averaged evaluation performance across all 10 folds of a specific model and a specific feature set is considered as the performance of that model and feature set pair.

With both of the reduced datasets, 10-fold validation was not performed, partly because we found that the standard errors of resulting accuracy between folds in the primary dataset evaluation to be low [135]. Instead, for each of the feature sets and each of the neural network configurations, a corresponding model is trained with 80% of the dataset for 300 epochs at most.

At the end of each epoch, the accuracy of the model is tested with 10% of the data – if there has been no improvements to the prediction accuracy resulting from this test set for 30 consecutive epochs, the training process is stopped early (i.e. early stopping); and the best performing state of the model, as tested with the test set at the end of each epoch, is considered as the trained model used for evaluation. Finally, the trained model is evaluated with the remaining 10% of the data that were not used for training nor end-of-epoch testing. The evaluation result is then considered as the performance of the feature set and model configuration pair [135].

Chapter 5

Results and Discussion

In this chapter, we present and discuss the results of two experiments. First, we consider the results from "Assessing the Use of Secondary Structure Fingerprints and Deep Learning to Classify RNA Sequences" [134]; see Section 5.1. Next, we discuss the results from "Extracting and Evaluating Features from RNA Virus Sequences to Predict Host Species Susceptibility Using Deep Learning" [135]; see Section 5.2.

5.1 Non-Coding RNAs from Rfam 14.1

This section begins with an overview of the classification performance of each feature set used in our study [134]. The work involves non-coding RNAs from the Rfam database [76,77]. Then, the section continues with analysis and discussion of the results.

5.1.1 Results Overview

Table 5.1, Table 5.2, and Table 5.3 show the 10-fold cross validation overall accuracy, precision, and recall respectively [134].

In this study, all of the structural fingerprints based on the common secondary structure motifs only used the match with the minimum free energy in case of multiple matches [134], as previously mentioned in Section 3.2. The micro averages are used in order to address class imbalances in the dataset (see Section 2.4.4).

Table 5.1: 10-fold validation accuracy and standard errors of the RNA classification problem using Rfam 14.1 [76, 77] dataset [134].

K-mer	RAG-based Fingerprints	Common Structure Fingerprints	Accuracy
-	✓	-	$32.94\% \pm 0.74\%$
-	-	✓(without wobble)	$52.28\% \pm 0.53\%$
-	-	✓(with wobble)	$55.43\% \pm 0.72\%$
-	-	✓(with and without wobble)	$60.36\% \pm 0.45\%$
-	✓	✓(with and without wobble)	$64.06\% \pm 0.99\%$
2-mer	-	-	$47.94\% \pm 0.48\%$
3-mer	-	-	$68.75\% \pm 0.54\%$
4-mer	-	-	$\mathbf{86.92\% \pm 0.14\%}$
2-mer	✓	✓(with and without wobble)	$75.23\% \pm 0.33\%$
3-mer	✓	✓(with and without wobble)	$81.61\% \pm 0.26\%$
4-mer	✓	✓(with and without wobble)	$85.49\% \pm 0.73\%$

Table 5.2: 10-fold validation precision and standard errors of the RNA classification problem using Rfam 14.1 [76, 77] dataset [134].

K-mer	RAG-based Fingerprints	Common Structure Fingerprints	Precision
-	✓	-	$51.30\% \pm 0.60\%$
-	-	✓(without wobble)	$68.70\% \pm 0.21\%$
-	-	✓(with wobble)	$70.70\% \pm 0.30\%$
-	-	✓(with and without wobble)	$72.30\% \pm 0.30\%$
-	✓	✓(with and without wobble)	$73.10\% \pm 0.23\%$
2-mer	-	-	$68.20\% \pm 0.29\%$
3-mer	-	-	$78.20\% \pm 0.25\%$
4-mer	-	-	$86.90\% \pm 0.10\%$
2-mer	✓	✓(with and without wobble)	$80.70\% \pm 0.21\%$
3-mer	✓	✓(with and without wobble)	$84.20\% \pm 0.13\%$
4-mer	✓	✓(with and without wobble)	$\mathbf{87.10\% \pm 0.23\%}$

Table 5.3: 10-fold validation recall and standard errors of the RNA classification problem using Rfam 14.1 [76, 77] dataset [134].

K-mer	RAG-based Fingerprints	Common Structure Fingerprints	Recall
-	✓	-	$33.00\% \pm 0.76\%$
-	-	✓(without wobble)	$52.40\% \pm 0.56\%$
-	-	✓(with wobble)	$55.60\% \pm 0.75\%$
-	-	✓(with and without wobble)	$60.40\% \pm 0.45\%$
-	✓	✓(with and without wobble)	$63.90\% \pm 0.98\%$
2-mer	-	-	$48.00\% \pm 0.47\%$
3-mer	-	-	$68.80\% \pm 0.55\%$
4-mer	-	-	$\mathbf{86.80\% \pm 0.20\%}$
2-mer	✓	✓(with and without wobble)	$75.20\% \pm 0.36\%$
3-mer	✓	✓(with and without wobble)	$81.70\% \pm 0.30\%$
4-mer	✓	✓(with and without wobble)	$85.40\% \pm 0.70\%$

5.1.2 Comparison of the Different Secondary Structure Fingerprint Types

When used on their own, the RAG-based secondary structure fingerprints performed worst at a 10-fold cross-validated accuracy of 32.94%; followed by the common secondary structure motifs based fingerprints that do not allow wobble pairs at 52.28%, then their counterparts that allow wobble pairs at 55.43% [134]. In terms of precision and recall, we observed a similar trend. The RAG-based fingerprints achieved the lowest precision and recall at 51.30% and 33.0% respectively, followed by the common structural motif fingerprints without wobble pairs at 68.70% and 52.40%, while the common motif fingerprints that allow wobble pairs achieved higher precision and recall at 70.70% and 55.60%.

There are several possible explanations behind the different overall classification performance between the RAG-based fingerprints and the fingerprints based on our curated common RNA secondary structure motifs [134]: First, it is possible that the RAG-based higher level topological structures are less useful to be used as fingerprints, compared to the curated smaller secondary structure motifs that focus on local structures. Second, it is also possible that the scores based on free energy values used for the fingerprints based on the curated common RNA secondary structure motifs, as opposed to scores based on the pseudoknots-compatible "bits" scoring scheme of RNAMotif [96] used to form the RAG-based fingerprints, provide more distinguishing information about the different classes of RNA. A third possibility is that using the score of the best match in case of multiple matches builds better and more informational fingerprints (which is how multiple matches were handled to build the curated common secondary structure motifs based fingerprints), compared to averaging the different scores of the multiple matches (how multiple matches were handled when building the RAG-based secondary structure fingerprints). Further experiments would be required to rule out the possible explanations that do not apply, and determine the actual cause behind the performance difference of the different secondary structure fingerprint types. Identifying the actual explanation/cause through further experiments would then allow future related studies to build better performing RNA secondary structure fingerprints.

When it comes to the common secondary structure motifs based fingerprints, allowing wobble pairs resulted in a slightly better classification performance than when wobble pairs are not allowed [134]. However, despite the poorer classification performance of the fingerprints that do not allow wobble pairs, their usefulness can still be observed when they are combined with the other fingerprints, as discussed in the following subsection.

5.1.3 Combining Different Secondary Structure Features

We observed improvements in overall classification accuracy when different types of secondary structure fingerprints are used together – i.e. the different sets of scores corresponding to the

different types of secondary structure fingerprints are concatenated, then used together as the input vector to the deep neural network. For instance, using the fingerprints based on the curated common known secondary structure motifs that allow and disallow G·U wobble pairs at the same time resulted in an improved 10-fold overall classification accuracy of 60.36%; from an overall accuracy of 55.43% when only the corresponding fingerprints that allow wobble pairs are used alone, and 52.28% when only the non-wobble pairs counterparts are used [134]. Combining the RAG-based fingerprints with this combination resulted in a further improved evaluation accuracy of 64.06% [134]. In other words, the three different types of secondary structure fingerprints complement each other, suggesting that each type contains certain information about the RNA class that is not in the other two types of the secondary structure fingerprints. Thus, future studies which involve building secondary structure fingerprints for RNA classification purposes should consider combining such different types to build a better performing set of fingerprints.

When using all three types of the secondary structure fingerprints combined, we noticed a trend on the classification performance – certain classes of RNAs are better predicted than others across the different folds. This trend is illustrated by the confusion matrices in Figure 5.1, Figure 5.2, and Figure 5.3.

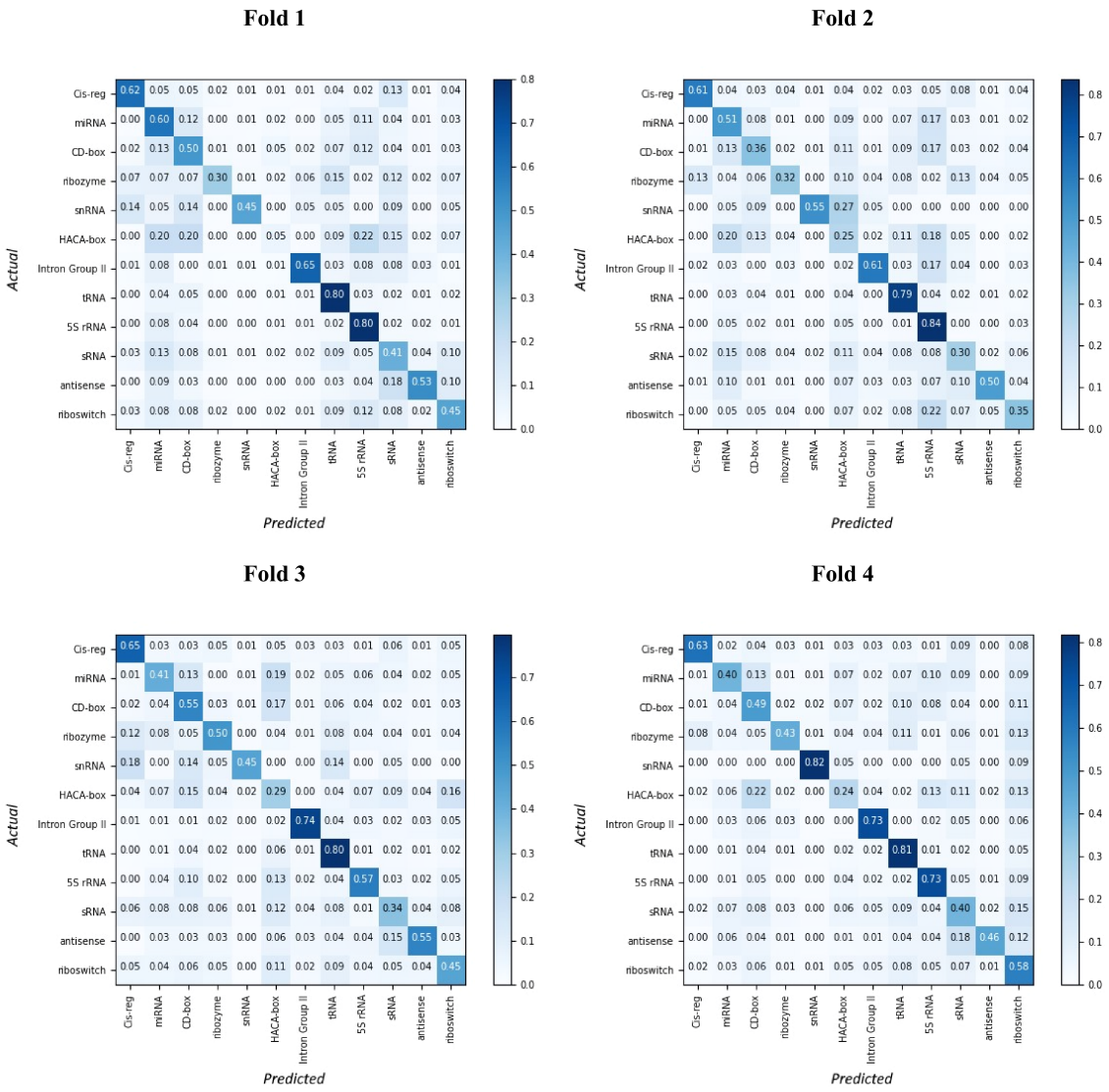

Figure 5.1: Resulting confusion matrices of fold 1 to 4 when all of the secondary structure fingerprints are used together as deep learning features for the RNA classification problem of the Rfam 14.1 [76, 77] dataset.

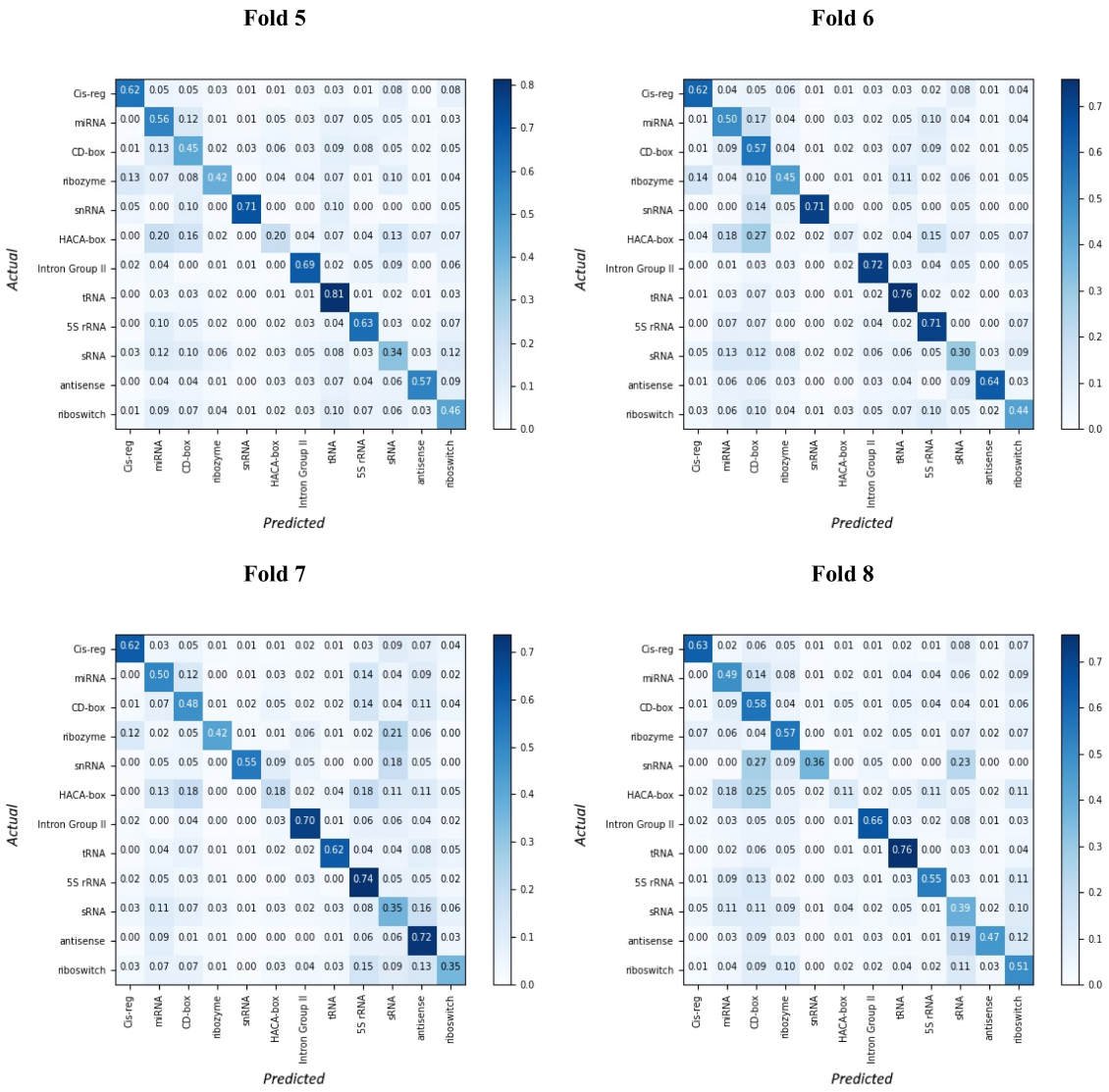

Figure 5.2: Resulting confusion matrices of fold 5 to 8 when all of the secondary structure fingerprints are used together as deep learning features for the RNA classification problem of the Rfam 14.1 [76, 77] dataset.

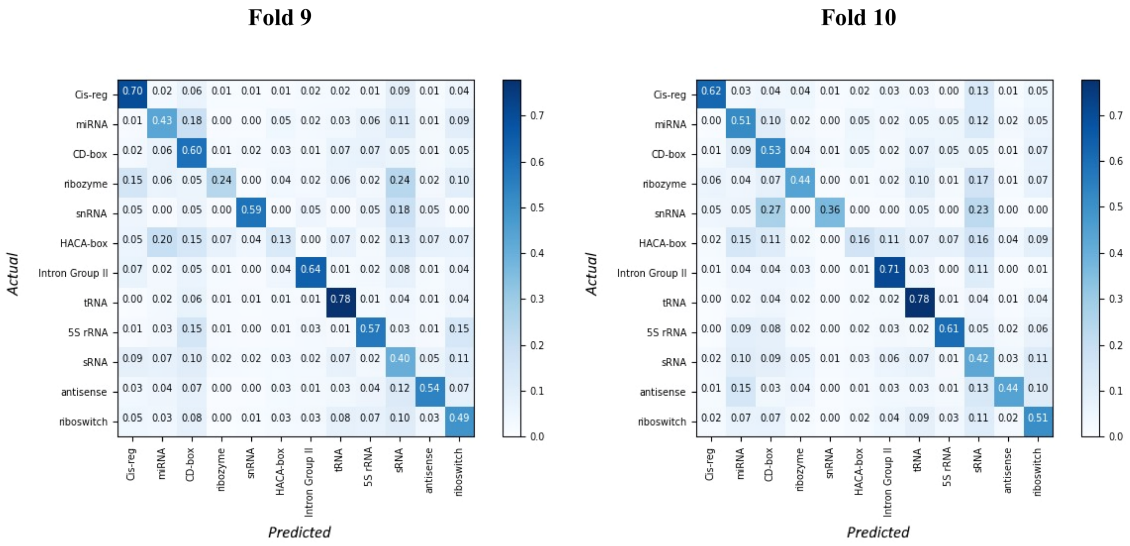

Figure 5.3: Resulting confusion matrices of fold 9 to 10 when all of the secondary structure fingerprints are used together as deep learning features for the RNA classification problem of the Rfam 14.1 [76, 77] dataset.

For instance, tRNA, which is one of the better predicted RNA classes using the secondary structure fingerprints alone without any sequence information, is known to form specific secondary structures (i.e. the "clover leaf" structure) on which its function strongly depends [141]. 5S rRNA, another better predicted RNA class, is also known to form rather consistent secondary structures, even in different organisms [11, 64]. When it comes to Intron Group II, structures are known to be more conserved than the sequences themselves [121], which explains why it is also one of the better predicted RNA classes. Finally, RNAs in the Cis-reg class are also known to involve secondary structures, such as hairpins [137], in their lifecycle and functions [70, 137].

Meanwhile, the prediction performance for antisense varied between the different folds, ranging from a rather poor classification result of 44% to a decent 72%, which may be due to the variation in secondary structures that can be formed by different "antisense RNAs" [148].

On the other hand, HACA-box was one of the poorly classified RNA classes using the secondary structure fingerprints as illustrated by the confusion matrices (Figure 5.1, Figure 5.2, and Figure 5.3). It was also one of the most misclassified RNA classes in a prior study that uses substructure information derived from predicted secondary structures with deep learning to classify RNAs [46]. The authors reasoned that this may be due to how HACA-box share the same "local

substructures" with other snoRNAs (e.g. CD-box) even when the global structures differ, and as a result, their substructure-based deep learning features resembling the "local substructures" provided limited useful infromation to the deep neural network to classify HACA-box correctly [46]. This may be the same reason why HACA-box was overall poorly classified in our study when all of the secondary structure fingerprints are used without any sequence-based features – as either local or global matches are allowed when the RNA sequences are searched for secondary structure matches in order to build the secondary structure fingerprints, and our current method does not distinguish between a local and global match.

These findings suggest that the usefulness of the secondary structure fingerprints would depend on the RNAs in question – if secondary structures are involved or conserved, then the fingerprints would serve as useful features for the deep neural network to make proper identification of the RNA class. Otherwise, additional features, such as sequence-based features, should be used in conjunction with or in place of the secondary structure features.

5.1.4 Sequence-Based Features and Secondary Structure Based Features

Overall, when it comes to using either sequence-based features (i.e. k-mers) or secondary structure based features separately on their own, using the former resulted in better classification performance compared to the latter [134]. However, with the exception of combining 4-mer with the secondary structure based features, using the sequence-based and the secondary structure based features together resulted in quite significant classification accuracy improvements – these improvements resulting from combining sequence-based features and secondary structure based features are more significant than the improvements that were observed when combining the different types of secondary structure fingerprints (i.e. no sequence-based features in the combination). For example, combining 2-mer with the all of the secondary structure fingerprints used in our study resulted in a superior classification accuracy of 75.23%, which is about 27% higher compared to the classification accuracy resulting from using 2-mer, and about 11% higher relative to the classification accuracy of the secondary structure fingerprints [134]. Similarly, using 3-mer together with the secondary structure features improved the classification accuracy to 81.61%; which is about 13% higher than the resulting classification accuracy of 3-mer, and about 18% higher than the resulting accuracy of the structural fingerprints [134].

However, the case is different when 4-mer is used together with the secondary structure fingerprints. Instead of improvements, we observed slight deterioration in the resulting averaged 10-fold classification accuracy – 4-mer alone without secondary structure related features produced the highest classification accuracy in our study at 86.92%, while having the secondary structure fingerprints also contribute to the classification in addition to 4-mer resulted in a slightly lower resulting classification accuracy at 85.49% [134]. When the comparison is made per fold, there

are only 3 out of the 10 folds in which slightly higher accuracy can be observed when the secondary structure fingerprints are used with the 4-mer. In other words, in 7 out of the 10 folds, or in the majority of the folds, including the secondary structure fingerprints deteriorated the classification accuracy of the 4-mer. It is possible that such classification accuracy deterioration is because the features combined have more noise compared to the 4-mer alone [124]. When it comes to precision, unlike the deterioration in accuracy, using the 4-mer together with the structural fingerprints resulted in a very small increase of precision of 0.20% – the average classification precision of 4-mer without the secondary structure fingerprints is 86.90%, while the average precision when the structural fingerprints are included is 87.10% [134]. However, this increase is small; and in terms of recall, we observed a larger decrease of $> 1\%$ similar to accuracy instead of an increase, from 86.80% to 85.40% [134]. Thus, to conclude the comparison between using 4-mer alone and 4-mer combined with the secondary structure fingerprints used, the combined features performed poorer than 4-mer on its own.

As increasing the value of k for the k-mers resulted in significant improvements in accuracy, precision, and recall in our study; it is likely that increasing k further would result in additional improvements for this classification problem [134]. However, one disadvantage of this is the "large dimensionality of the features" [134], and as a result, the larger required computational resources to train and evaluate the deep neural networks. For example, 4-mer consists of 256 values/features, whereas 3-mer combined with the secondary structure fingerprints used in the study consist of 199 values/features in total [134].

5.2 RNA Virus Sequences and their Host Species from NCBI Virus

The results we obtained in our study involving identification of susceptible hosts given sequences of RNA viruses [135] are presented in the beginning of this section. These are then followed by our analyses and discussion.

5.2.1 Primary Results Overview

Table 5.4, Table 5.5, and Table 5.6 show the 10-fold validation overall micro averaged accuracy, precision, and recall respectively (along with the standard errors resulting from the differences in accuracy, precision, and recall respectively between the 10 different folds) for each of the feature set used in the study [135].

The common secondary structure motifs based fingerprints in this study do not allow wobble pairs [135].

5.2.2 Reduced Dataset Results Overview

Table 5.7 shows the resulting test set accuracy for each of the feature sets, using the reduced dataset in which sequences are 90% or less in similarity to each other; while Table 5.8 shows the results when the second reduced dataset, with only sequences that are 80% or less in similarity, is used [135].

Table 5.4: 10-fold validation accuracy and standard errors of the RNA virus host classification problem, with the highest accuracy of each feature set bolded, and overall highest underlined [135].

K-mer	"Skip-mer" [31]			Common Structure Fingerprints	10-Fold Cross Validation Averaged Accuracy		
	Length	Match	Skip		3-Layers Model	4-Layers Model	5-Layers Model
4-mer		-		-	62.48% ± 0.51%	**64.86%** ± 0.76%	62.09% ± 0.77%
5-mer		-		-	**77.29%** ± 0.22%	75.24% ± 0.53%	74.31% ± 0.46%
6-mer		-		-	**84.56%** ± 0.28%	83.55% ± 0.48%	83.55% ± 0.57%
-	6	2	1	-	61.74% ± 0.31%	**61.85%** ± 0.94%	59.45% ± 1.0%
-	7	1	1	-	**55.89%** ± 0.34%	54.38% ± 0.99%	48.39% ± 1.86%
-	7	2	1	-	**77.32%** ± 0.5%	75.76% ± 0.8%	71.74% ± 1.83%
-	9	1	1	-	**75.16%** ± 0.41%	73.23% ± 0.46%	65.53% ± 4.57%
-	9	2	1	-	**84.92%** ± 0.25%	84.0% ± 0.36%	82.2% ± 1.13%
-	11	1	1	-	**84.08%** ± 0.21%	81.78% ± 0.98%	81.15% ± 0.88%
-		-		min. free energy	35.91% ± 0.42%	**36.75%** ± 0.76%	35.94% ± 0.51%
-		-		min., avg. free energy	50.65% ± 0.57%	52.04% ± 0.86%	**52.6%** ± 0.65%
-		-		min., avg., max. free energy	57.37% ± 0.52%	59.39% ± 0.58%	**59.42%** ± 0.76%
4-mer	6	2	1	-	71.57% ± 0.4%	**71.69%** ± 0.41%	71.15% ± 0.49%
4-mer	7	1	1	-	70.52% ± 0.39%	**71.91%** ± 0.38%	69.63% ± 1.01%
5-mer	7	2	1	-	**82.14%** ± 0.29%	82.08% ± 0.46%	80.1% ± 0.65%
5-mer	9	1	1	-	**81.77%** ± 0.47%	81.13% ± 0.39%	80.39% ± 0.69%
6-mer	9	2	1	-	**86.89%** ± 0.28%	86.09% ± 0.21%	84.68% ± 0.83%
6-mer	11	1	1	-	**86.7%** ± 0.38%	86.17% ± 0.61%	84.73% ± 1.58%
4-mer		-		min. free energy	67.96% ± 0.56%	70.97% ± 0.54%	**72.6%** ± 0.63%
5-mer		-		min. free energy	78.93% ± 0.24%	80.49% ± 0.62%	**81.05%** ± 0.46%
6-mer		-		min. free energy	**84.33%** ± 0.51%	84.05% ± 0.71%	77.7% ± 5.36%
4-mer		-		min., avg. free energy	69.92% ± 0.56%	72.28% ± 0.52%	**75.38%** ± 0.6%
5-mer		-		min., avg. free energy	74.93% ± 1.66%	**81.28%** ± 0.43%	81.02% ± 0.33%
6-mer		-		min., avg. free energy	83.42% ± 0.39%	**83.73%** ± 0.32%	82.23% ± 0.32%
4-mer		-		min., avg., max. free energy	71.14% ± 0.49%	74.63% ± 0.54%	**75.85%** ± 0.54%
5-mer		-		min., avg., max. free energy	79.28% ± 0.75%	80.74% ± 0.52%	**81.23%** ± 0.75%
6-mer		-		min., avg., max. free energy	83.21% ± 0.37%	**83.53%** ± 0.13%	81.87% ± 0.44%
-	6	2	1	min. free energy	66.94% ± 0.58%	69.98% ± 0.65%	**71.02%** ± 0.83%
-	7	1	1	min. free energy	66.83% ± 0.22%	69.69% ± 0.48%	**71.23%** ± 0.34%
-	7	2	1	min. free energy	78.66% ± 0.55%	80.35% ± 0.41%	**80.72%** ± 0.59%
-	9	1	1	min. free energy	77.78% ± 0.27%	**80.05%** ± 0.29%	79.43% ± 1.73%
-	9	2	1	min. free energy	**84.58%** ± 0.34%	79.17% ± 3.18%	80.65% ± 1.28%
-	11	1	1	min. free energy	83.61% ± 0.52%	**83.88%** ± 0.32%	77.77% ± 5.17%
-	6	2	1	min., avg. free energy	69.62% ± 0.49%	71.83% ± 0.74%	**74.16%** ± 0.6%
-	7	1	1	min., avg. free energy	68.11% ± 0.99%	71.45% ± 0.73%	**73.93%** ± 0.59%
-	7	2	1	min., avg. free energy	78.64% ± 0.35%	79.75% ± 0.85%	**80.9%** ± 0.32%
-	9	1	1	min., avg. free energy	78.48% ± 0.59%	79.58% ± 0.55%	**81.29%** ± 0.32%
-	9	2	1	min., avg. free energy	83.2% ± 0.54%	**83.37%** ± 0.43%	82.45% ± 0.57%
-	11	1	1	min., avg. free energy	**82.83%** ± 0.41%	82.75% ± 0.44%	82.3% ± 0.45%
-	6	2	1	min., avg., max. free energy	70.77% ± 0.42%	74.04% ± 0.75%	**75.38%** ± 0.36%
-	7	1	1	min., avg., max. free energy	69.28% ± 0.75%	74.32% ± 0.52%	**74.74%** ± 0.75%
-	7	2	1	min., avg., max. free energy	79.02% ± 0.58%	80.43% ± 0.52%	**81.16%** ± 0.52%
-	9	1	1	min., avg., max. free energy	78.73% ± 0.45%	80.42% ± 0.7%	**81.3%** ± 0.33%
-	9	2	1	min., avg., max. free energy	**83.93%** ± 0.2%	83.38% ± 0.42%	82.34% ± 0.63%
-	11	1	1	min., avg., max. free energy	**83.04%** ± 0.4%	83.0% ± 0.18%	82.47% ± 0.35%
4-mer	6	2	1	min. free energy	74.26% ± 0.46%	76.11% ± 0.66%	**78.78%** ± 0.3%
4-mer	7	1	1	min. free energy	74.22% ± 0.29%	77.44% ± 0.83%	**78.54%** ± 0.47%
5-mer	7	2	1	min. free energy	**83.74%** ± 0.39%	83.54% ± 0.39%	83.65% ± 0.18%
5-mer	9	1	1	min. free energy	82.17% ± 0.47%	**83.21%** ± 0.42%	83.07% ± 0.45%
6-mer	9	2	1	min. free energy	**85.9%** ± 0.28%	84.37% ± 0.71%	83.1% ± 0.73%
6-mer	11	1	1	min. free energy	**85.86%** ± 0.35%	84.94% ± 0.34%	82.39% ± 0.57%
4-mer	6	2	1	min., avg. free energy	75.06% ± 0.53%	77.33% ± 0.66%	**78.89%** ± 0.48%
4-mer	7	1	1	min., avg. free energy	74.78% ± 0.46%	77.11% ± 0.39%	**78.47%** ± 0.48%
5-mer	7	2	1	min., avg. free energy	82.52% ± 0.38%	**82.77%** ± 0.41%	82.68% ± 0.27%
5-mer	9	1	1	min., avg. free energy	81.26% ± 0.38%	**82.59%** ± 0.6%	82.37% ± 0.41%
6-mer	9	2	1	min., avg. free energy	**84.39%** ± 0.52%	84.2% ± 0.3%	82.54% ± 1.01%
6-mer	11	1	1	min., avg. free energy	**84.33%** ± 0.53%	84.19% ± 0.65%	82.06% ± 0.7%
4-mer	6	2	1	min., avg., max. free energy	75.73% ± 0.57%	79.56% ± 0.2%	**79.65%** ± 0.67%
4-mer	7	1	1	min., avg., max. free energy	75.77% ± 0.61%	77.99% ± 0.38%	**79.23%** ± 0.44%
5-mer	7	2	1	min., avg., max. free energy	82.54% ± 0.39%	**82.92%** ± 0.28%	82.16% ± 0.61%
5-mer	9	1	1	min., avg., max. free energy	81.41% ± 0.34%	**83.13%** ± 0.17%	81.55% ± 1.31%
6-mer	9	2	1	min., avg., max. free energy	**84.73%** ± 0.32%	83.71% ± 0.16%	82.33% ± 0.64%
6-mer	11	1	1	min., avg., max. free energy	**84.61%** ± 0.26%	83.33% ± 0.68%	80.4% ± 2.18%

Table 5.5: 10-fold validation precision and standard errors of the RNA virus host classification problem, with the highest precision of each feature set bolded, and overall highest underlined.

K-mer	"Skip-mer" [31]			Common Structure Fingerprints	10-Fold Cross Validation Averaged Precision		
	Length	Match	Skip		3-Layers Model	4-Layers Model	5-Layers Model
4-mer		-		-	**79.8%** ± 0.2%	79.7% ± 0.3%	78.1% ± 0.43%
5-mer		-		-	**84.4%** ± 0.16%	83.5% ± 0.27%	83.0% ± 0.15%
6-mer		-		-	**87.1%** ± 0.23%	86.4% ± 0.22%	86.5% ± 0.27%
-	6	2	1	-	**79.9%** ± 0.23%	79.3% ± 0.3%	78.8% ± 0.33%
-	7	1	1	-	**77.9%** ± 0.1%	77.6% ± 0.27%	75.5% ± 0.56%
-	7	2	1	-	**84.1%** ± 0.23%	83.5% ± 0.22%	82.2% ± 0.51%
-	9	1	1	-	**83.1%** ± 0.18%	82.2% ± 0.13%	80.1% ± 0.74%
-	9	2	1	-	**87.4%** ± 0.16%	86.9% ± 0.18%	85.9% ± 0.35%
-	11	1	1	-	**86.6%** ± 0.16%	85.6% ± 0.37%	85.3% ± 0.42%
-		-		min. free energy	69.1% ± 0.28%	**69.7%** ± 0.26%	69.0% ± 0.3%
-		-		min., avg. free energy	74.5% ± 0.22%	**75.2%** ± 0.29%	75.1% ± 0.31%
-		-		min., avg., max. free energy	76.1% ± 0.18%	**77.1%** ± 0.18%	76.8% ± 0.29%
4-mer	6	2	1	-	**82.7%** ± 0.15%	82.4% ± 0.16%	82.1% ± 0.23%
4-mer	7	1	1	-	**82.4%** ± 0.22%	82.1% ± 0.18%	81.8% ± 0.25%
5-mer	7	2	1	-	**86.3%** ± 0.15%	85.9% ± 0.18%	85.5% ± 0.22%
5-mer	9	1	1	-	**85.9%** ± 0.18%	85.6% ± 0.22%	85.4% ± 0.22%
6-mer	9	2	1	-	**88.4%** ± 0.16%	88.2% ± 0.13%	87.3% ± 0.33%
6-mer	11	1	1	-	**88.6% ± 0.16%**	87.9% ± 0.31%	87.3% ± 0.5%
4-mer		-		min. free energy	80.7% ± 0.21%	81.4% ± 0.22%	**81.6%** ± 0.16%
5-mer		-		min. free energy	84.4% ± 0.27%	**84.7%** ± 0.37%	84.6% ± 0.27%
6-mer		-		min. free energy	**87.1%** ± 0.23%	86.4% ± 0.27%	84.4% ± 1.19%
4-mer		-		min., avg. free energy	81.3% ± 0.21%	81.8% ± 0.2%	**82.1%** ± 0.18%
5-mer		-		min., avg. free energy	84.0% ± 0.33%	**84.9%** ± 0.23%	84.3% ± 0.15%
6-mer		-		min., avg. free energy	**86.5%** ± 0.22%	85.8% ± 0.13%	85.0% ± 0.15%
4-mer		-		min., avg., max. free energy	81.5% ± 0.17%	82.1% ± 0.18%	**82.2%** ± 0.2%
5-mer		-		min., avg., max. free energy	84.6% ± 0.27%	**84.8%** ± 0.2%	84.5% ± 0.27%
6-mer		-		min., avg., max. free energy	**86.2%** ± 0.2%	85.6% ± 0.16%	84.3% ± 0.26%
-	6	2	1	min. free energy	80.0% ± 0.3%	80.8% ± 0.13%	**81.1%** ± 0.23%
-	7	1	1	min. free energy	79.6% ± 0.16%	80.5% ± 0.17%	**80.6%** ± 0.16%
-	7	2	1	min. free energy	84.3% ± 0.26%	84.5% ± 0.22%	**84.6%** ± 0.22%
-	9	1	1	min. free energy	83.8% ± 0.13%	**84.4%** ± 0.16%	83.6% ± 0.45%
-	9	2	1	min. free energy	**87.2%** ± 0.2%	84.9% ± 0.99%	85.3% ± 0.42%
-	11	1	1	min. free energy	**87.2%** ± 0.2%	86.2% ± 0.25%	82.6% ± 2.75%
-	6	2	1	min., avg. free energy	81.0% ± 0.21%	**81.7%** ± 0.21%	**81.7%** ± 0.21%
-	7	1	1	min., avg. free energy	80.2% ± 0.2%	81.0% ± 0.15%	**81.6%** ± 0.22%
-	7	2	1	min., avg. free energy	84.1% ± 0.18%	**84.6%** ± 0.31%	84.4% ± 0.16%
-	9	1	1	min., avg. free energy	84.0% ± 0.21%	84.1% ± 0.18%	**84.4%** ± 0.16%
-	9	2	1	min., avg. free energy	**86.5%** ± 0.22%	85.8% ± 0.2%	85.1% ± 0.23%
-	11	1	1	min., avg. free energy	**86.3%** ± 0.15%	85.3% ± 0.21%	84.9% ± 0.23%
-	6	2	1	min., avg., max. free energy	81.4% ± 0.16%	82.1% ± 0.18%	**82.4%** ± 0.16%
-	7	1	1	min., avg., max. free energy	80.6% ± 0.22%	**81.9%** ± 0.23%	81.7% ± 0.15%
-	7	2	1	min., avg., max. free energy	84.4% ± 0.16%	**84.5%** ± 0.22%	**84.5%** ± 0.22%
-	9	1	1	min., avg., max. free energy	**84.4%** ± 0.16%	84.3% ± 0.21%	84.2% ± 0.25%
-	9	2	1	min., avg., max. free energy	**86.7%** ± 0.15%	85.6% ± 0.27%	84.7% ± 0.37%
-	11	1	1	min., avg., max. free energy	**86.2%** ± 0.2%	85.4% ± 0.16%	84.4% ± 0.22%
4-mer	6	2	1	min. free energy	82.9% ± 0.18%	83.4% ± 0.16%	**83.5%** ± 0.17%
4-mer	7	1	1	min. free energy	82.7% ± 0.21%	**83.6%** ± 0.22%	**83.6%** ± 0.16%
5-mer	7	2	1	min. free energy	**86.8%** ± 0.2%	86.3% ± 0.15%	85.9% ± 0.18%
5-mer	9	1	1	min. free energy	**86.1%** ± 0.23%	**86.1%** ± 0.28%	85.6% ± 0.16%
6-mer	9	2	1	min. free energy	**87.9%** ± 0.1%	86.9% ± 0.28%	86.4% ± 0.31%
6-mer	11	1	1	min. free energy	**88.0%** ± 0.21%	87.0% ± 0.21%	86.1% ± 0.18%
4-mer	6	2	1	min., avg. free energy	82.9% ± 0.18%	83.3% ± 0.21%	**83.8%** ± 0.13%
4-mer	7	1	1	min., avg. free energy	83.2% ± 0.2%	83.3% ± 0.15%	**83.6%** ± 0.22%
5-mer	7	2	1	min., avg. free energy	**86.2%** ± 0.25%	85.6% ± 0.16%	85.2% ± 0.2%
5-mer	9	1	1	min., avg. free energy	85.7% ± 0.15%	**86.1%** ± 0.23%	84.8% ± 0.2%
6-mer	9	2	1	min., avg. free energy	**87.4%** ± 0.16%	86.7% ± 0.21%	85.8% ± 0.36%
6-mer	11	1	1	min., avg. free energy	**87.6%** ± 0.16%	86.3% ± 0.33%	85.4% ± 0.31%
4-mer	6	2	1	min., avg., max. free energy	82.9% ± 0.23%	**84.2%** ± 0.2%	83.9% ± 0.31%
4-mer	7	1	1	min., avg., max. free energy	82.9% ± 0.18%	83.8% ± 0.25%	**83.8%** ± 0.2%
5-mer	7	2	1	min., avg., max. free energy	**86.0%** ± 0.21%	85.7% ± 0.15%	84.8% ± 0.2%
5-mer	9	1	1	min., avg., max. free energy	**85.8%** ± 0.2%	85.4% ± 0.16%	84.5% ± 0.5%
6-mer	9	2	1	min., avg., max. free energy	**86.9%** ± 0.18%	86.2% ± 0.13%	85.4% ± 0.27%
6-mer	11	1	1	min., avg., max. free energy	**87.0%** ± 0.15%	85.8% ± 0.29%	84.5% ± 0.75%

Table 5.6: 10-fold validation recall and standard errors of the RNA virus host classification problem, with the highest recall of each feature set bolded, and overall highest underlined.

K-mer	"Skip-mer" [31]			Common Structure Fingerprints	10-Fold Cross Validation Averaged Recall		
	Length	Match	Skip		3-Layers Model	4-Layers Model	5-Layers Model
4-mer		-		-	62.5% ± 0.48%	**64.7%** ± 0.73%	62.2% ± 0.87%
5-mer		-		-	**77.3%** ± 0.26%	75.3% ± 0.54%	74.4% ± 0.5%
6-mer		-		-	**84.6%** ± 0.34%	83.6% ± 0.5%	83.6% ± 0.62%
-	6	2	1	-	61.7% ± 0.3%	**61.9%** ± 0.96%	59.4% ± 1.03%
-	7	1	1	-	**55.8%** ± 0.33%	54.5% ± 0.96%	48.5% ± 1.83%
-	7	2	1	-	**77.3%** ± 0.52%	75.6% ± 0.81%	71.7% ± 1.78%
-	9	1	1	-	**75.0%** ± 0.39%	73.1% ± 0.48%	65.5% ± 4.55%
-	9	2	1	-	**84.9%** ± 0.23%	84.1% ± 0.43%	82.2% ± 1.19%
-	11	1	1	-	**84.1%** ± 0.23%	81.7% ± 1.04%	81.2% ± 0.9%
-		-		min. free energy	35.9% ± 0.48%	**36.6%** ± 0.72%	35.9% ± 0.53%
-		-		min., avg. free energy	50.5% ± 0.54%	52.0% ± 0.87%	**52.6%** ± 0.58%
-		-		min., avg., max. free energy	57.4% ± 0.52%	59.4% ± 0.6%	**59.5%** ± 0.79%
4-mer	6	2	1	-	71.5% ± 0.4%	**71.7%** ± 0.42%	71.1% ± 0.53%
4-mer	7	1	1	-	70.5% ± 0.43%	**71.9%** ± 0.41%	69.5% ± 1.05%
5-mer	7	2	1	-	**82.2%** ± 0.25%	82.1% ± 0.55%	80.0% ± 0.68%
5-mer	9	1	1	-	**81.7%** ± 0.45%	81.1% ± 0.38%	80.3% ± 0.65%
6-mer	9	2	1	-	**87.0%** ± 0.3%	86.2% ± 0.25%	84.7% ± 0.84%
6-mer	11	1	1	-	**86.8%** ± 0.44%	86.3% ± 0.58%	84.7% ± 1.54%
4-mer		-		min. free energy	68.0% ± 0.58%	71.0% ± 0.6%	**72.7%** ± 0.6%
5-mer		-		min. free energy	78.8% ± 0.25%	80.3% ± 0.6%	**81.0%** ± 0.52%
6-mer		-		min. free energy	**84.4%** ± 0.54%	84.1% ± 0.74%	77.8% ± 5.35%
4-mer		-		min., avg. free energy	70.1% ± 0.57%	72.3% ± 0.54%	**75.3%** ± 0.6%
5-mer		-		min., avg. free energy	74.9% ± 1.61%	**81.3%** ± 0.42%	81.0% ± 0.33%
6-mer		-		min., avg. free energy	83.5% ± 0.43%	**83.7%** ± 0.33%	82.3% ± 0.37%
4-mer		-		min., avg., max. free energy	71.2% ± 0.47%	74.6% ± 0.52%	**76.0%** ± 0.56%
5-mer		-		min., avg., max. free energy	79.2% ± 0.73%	80.7% ± 0.5%	**81.2%** ± 0.73%
6-mer		-		min., avg., max. free energy	83.3% ± 0.4%	**83.6%** ± 0.16%	82.0% ± 0.47%
-	6	2	1	min. free energy	67.0% ± 0.58%	70.0% ± 0.73%	**71.2%** ± 0.85%
-	7	1	1	min. free energy	66.9% ± 0.23%	69.7% ± 0.52%	**71.1%** ± 0.31%
-	7	2	1	min. free energy	78.7% ± 0.6%	80.4% ± 0.37%	**80.8%** ± 0.55%
-	9	1	1	min. free energy	77.8% ± 0.25%	**79.9%** ± 0.31%	79.4% ± 1.73%
-	9	2	1	min. free energy	**84.8%** ± 0.36%	79.1% ± 3.21%	80.7% ± 1.25%
-	11	1	1	min. free energy	83.7% ± 0.54%	**83.8%** ± 0.33%	77.8% ± 5.17%
-	6	2	1	min., avg. free energy	69.8% ± 0.51%	71.8% ± 0.73%	**74.2%** ± 0.55%
-	7	1	1	min., avg. free energy	68.1% ± 0.97%	71.5% ± 0.73%	**74.0%** ± 0.68%
-	7	2	1	min., avg. free energy	78.7% ± 0.33%	79.8% ± 0.85%	**80.8%** ± 0.33%
-	9	1	1	min., avg. free energy	78.6% ± 0.58%	79.5% ± 0.56%	**81.3%** ± 0.33%
-	9	2	1	min., avg. free energy	83.2% ± 0.57%	**83.5%** ± 0.43%	82.6% ± 0.56%
-	11	1	1	min., avg. free energy	**82.8%** ± 0.42%	82.7% ± 0.42%	82.2% ± 0.44%
-	6	2	1	min., avg., max. free energy	70.9% ± 0.43%	73.9% ± 0.74%	**75.4%** ± 0.31%
-	7	1	1	min., avg., max. free energy	69.3% ± 0.79%	74.2% ± 0.53%	**74.7%** ± 0.76%
-	7	2	1	min., avg., max. free energy	79.0% ± 0.56%	80.6% ± 0.54%	**81.1%** ± 0.55%
-	9	1	1	min., avg., max. free energy	78.7% ± 0.47%	80.6% ± 0.7%	**81.3%** ± 0.37%
-	9	2	1	min., avg., max. free energy	**84.0%** ± 0.21%	83.4% ± 0.43%	82.2% ± 0.66%
-	11	1	1	min., avg., max. free energy	82.9% ± 0.38%	**83.2%** ± 0.2%	82.4% ± 0.37%
4-mer	6	2	1	min. free energy	74.2% ± 0.49%	76.2% ± 0.7%	**78.7%** ± 0.3%
4-mer	7	1	1	min. free energy	74.2% ± 0.29%	77.5% ± 0.83%	**78.5%** ± 0.52%
5-mer	7	2	1	min. free energy	83.7% ± 0.42%	83.5% ± 0.43%	**83.8%** ± 0.2%
5-mer	9	1	1	min. free energy	82.2% ± 0.47%	**83.4%** ± 0.48%	83.1% ± 0.46%
6-mer	9	2	1	min. free energy	**85.9%** ± 0.31%	84.3% ± 0.7%	83.2% ± 0.7%
6-mer	11	1	1	min. free energy	**85.7%** ± 0.4%	84.9% ± 0.31%	82.4% ± 0.58%
4-mer	6	2	1	min., avg. free energy	75.1% ± 0.5%	77.4% ± 0.65%	**78.9%** ± 0.48%
4-mer	7	1	1	min., avg. free energy	74.7% ± 0.45%	77.0% ± 0.42%	**78.4%** ± 0.5%
5-mer	7	2	1	min., avg. free energy	82.6% ± 0.4%	**82.9%** ± 0.41%	82.7% ± 0.3%
5-mer	9	1	1	min., avg. free energy	81.3% ± 0.37%	82.6% ± 0.62%	**82.6%** ± 0.37%
6-mer	9	2	1	min., avg. free energy	**84.4%** ± 0.52%	84.3% ± 0.26%	82.5% ± 1.02%
6-mer	11	1	1	min., avg. free energy	**84.3%** ± 0.5%	84.2% ± 0.66%	82.1% ± 0.74%
4-mer	6	2	1	min., avg., max. free energy	75.6% ± 0.58%	79.6% ± 0.22%	**79.7%** ± 0.67%
4-mer	7	1	1	min., avg., max. free energy	75.8% ± 0.57%	77.8% ± 0.42%	**79.3%** ± 0.45%
5-mer	7	2	1	min., avg., max. free energy	82.6% ± 0.4%	**83.0%** ± 0.3%	82.2% ± 0.61%
5-mer	9	1	1	min., avg., max. free energy	81.4% ± 0.34%	**83.0%** ± 0.21%	81.5% ± 1.33%
6-mer	9	2	1	min., avg., max. free energy	**84.6%** ± 0.34%	83.7% ± 0.15%	82.3% ± 0.68%
6-mer	11	1	1	min., avg., max. free energy	**84.6%** ± 0.22%	83.3% ± 0.68%	80.3% ± 2.15%

Table 5.7: Test set accuracy of each feature set for the RNA virus host classification problem, using a reduced dataset that excludes sequences with $> 90\%$ similarity [135].

K-mer	"Skip-mer" [31]			Common Structure Fingerprints	Test Set Accuracy		
	Length	Match	Skip		3-Layers Model	4-Layers Model	5-Layers Model
4-mer		-		-	52.72%	53.67%	**59.74%**
5-mer		-		-	**72.52%**	70.29%	70.61%
6-mer		-		-	**80.51%**	75.08%	71.57%
-	6	2	1	-	56.23%	62.3%	**65.81%**
-	7	1	1	-	41.21%	44.09%	**52.08%**
-	7	2	1	-	**75.4%**	71.25%	71.88%
-	9	1	1	-	48.24%	**64.54%**	61.98%
-	9	2	1	-	**79.87%**	71.25%	67.73%
-	11	1	1	-	**68.37%**	**68.37%**	66.45%
-		-		min. free energy	**39.94%**	38.02%	**39.94%**
-		-		min., avg. free energy	51.44%	48.88%	**59.11%**
-		-		min., avg., max. free energy	46.33%	48.56%	**48.88%**
4-mer	6	2	1	-	**71.25%**	70.93%	52.72%
4-mer	7	1	1	-	**72.2%**	69.33%	71.57%
5-mer	7	2	1	-	**78.27%**	77.0%	73.8%
5-mer	9	1	1	-	**69.97%**	65.81%	69.65%
6-mer	9	2	1	-	**76.36%**	71.88%	68.69%
6-mer	11	1	1	-	**72.84%**	71.57%	53.99%
4-mer		-		min. free energy	61.66%	**61.98%**	49.2%
5-mer		-		min. free energy	67.09%	**72.2%**	69.01%
6-mer		-		min. free energy	**72.2%**	69.97%	61.98%
4-mer		-		min., avg. free energy	63.58%	59.42%	**63.9%**
5-mer		-		min., avg. free energy	64.22%	**66.77%**	**66.77%**
6-mer		-		min., avg. free energy	**69.65%**	63.9%	58.47%
4-mer		-		min., avg., max. free energy	59.74%	60.06%	**65.18%**
5-mer		-		min., avg., max. free energy	**65.81%**	64.22%	56.23%
6-mer		-		min., avg., max. free energy	**64.86%**	63.26%	52.08%
-	6	2	1	min. free energy	60.7%	59.42%	**65.5%**
-	7	1	1	min. free energy	42.17%	**48.24%**	47.92%
-	7	2	1	min. free energy	60.06%	69.01%	**70.29%**
-	9	1	1	min. free energy	44.73%	47.6%	**63.26%**
-	9	2	1	min. free energy	68.69%	**69.01%**	68.05%
-	11	1	1	min. free energy	**71.25%**	68.37%	64.86%
-	6	2	1	min., avg. free energy	53.99%	**58.47%**	55.91%
-	7	1	1	min., avg. free energy	**54.31%**	53.04%	52.72%
-	7	2	1	min., avg. free energy	59.11%	67.41%	**69.33%**
-	9	1	1	min., avg. free energy	63.26%	61.66%	**64.86%**
-	9	2	1	min., avg. free energy	**71.57%**	63.58%	65.81%
-	11	1	1	min., avg. free energy	**69.33%**	66.13%	62.62%
-	6	2	1	min., avg., max. free energy	57.51%	57.19%	**59.74%**
-	7	1	1	min., avg., max. free energy	**53.35%**	**53.35%**	52.08%
-	7	2	1	min., avg., max. free energy	57.83%	**62.62%**	59.74%
-	9	1	1	min., avg., max. free energy	51.76%	63.26%	**65.81%**
-	9	2	1	min., avg., max. free energy	67.41%	**68.37%**	64.22%
-	11	1	1	min., avg., max. free energy	62.3%	**63.9%**	61.66%
4-mer	6	2	1	min. free energy	**67.09%**	64.22%	65.5%
4-mer	7	1	1	min. free energy	**60.7%**	59.74%	47.28%
5-mer	7	2	1	min. free energy	**77.0%**	72.2%	66.13%
5-mer	9	1	1	min. free energy	70.29%	**72.84%**	71.88%
6-mer	9	2	1	min. free energy	**76.04%**	68.69%	66.45%
6-mer	11	1	1	min. free energy	**78.59%**	77.32%	61.66%
4-mer	6	2	1	min., avg. free energy	60.06%	60.38%	**61.02%**
4-mer	7	1	1	min., avg. free energy	**66.77%**	66.45%	65.18%
5-mer	7	2	1	min., avg. free energy	67.73%	**68.37%**	63.58%
5-mer	9	1	1	min., avg. free energy	**70.61%**	66.77%	69.33%
6-mer	9	2	1	min., avg. free energy	**77.32%**	69.97%	63.58%
6-mer	11	1	1	min., avg. free energy	**73.48%**	69.65%	57.51%
4-mer	6	2	1	min., avg., max. free energy	67.73%	65.18%	**68.69%**
4-mer	7	1	1	min., avg., max. free energy	59.11%	**63.9%**	**63.9%**
5-mer	7	2	1	min., avg., max. free energy	**66.13%**	61.34%	**66.13%**
5-mer	9	1	1	min., avg., max. free energy	**71.88%**	68.37%	66.13%
6-mer	9	2	1	min., avg., max. free energy	**77.32%**	74.76%	59.42%
6-mer	11	1	1	min., avg., max. free energy	**67.09%**	62.3%	58.15%

Table 5.8: Test set accuracy of each feature set for the RNA virus host classification problem, using a reduced dataset that excludes sequences with $> 80\%$ similarity [135].

K-mer	"Skip-mer" [31]			Common Structure Fingerprints	Test Set Accuracy		
	Length	Match	Skip		3-Layers Model	4-Layers Model	5-Layers Model
4-mer		-		-	**44.22%**	41.5%	40.14%
5-mer		-		-	46.26%	**48.3%**	35.37%
6-mer		-		-	**53.74%**	47.62%	36.73%
-	6	2	1	-	31.97%	**46.26%**	42.86%
-	7	1	1	-	33.33%	**36.73%**	35.37%
-	7	2	1	-	**54.42%**	34.01%	36.73%
-	9	1	1	-	**42.18%**	35.37%	32.65%
-	9	2	1	-	46.26%	**49.66%**	19.05%
-	11	1	1	-	**47.62%**	26.53%	26.53%
-		-		min. free energy	25.17%	25.17%	**31.29%**
-		-		min., avg. free energy	31.29%	31.97%	**32.65%**
-		-		min., avg., max. free energy	35.37%	**36.05%**	33.33%
4-mer	6	2	1	-	45.58%	48.3%	**50.34%**
4-mer	7	1	1	-	**44.9%**	44.22%	43.54%
5-mer	7	2	1	-	**60.54%**	56.46%	39.46%
5-mer	9	1	1	-	**59.18%**	50.34%	50.34%
6-mer	9	2	1	-	**57.14%**	50.34%	42.18%
6-mer	11	1	1	-	**57.82%**	44.9%	42.86%
4-mer		-		min. free energy	30.61%	31.29%	**41.5%**
5-mer		-		min. free energy	44.22%	**46.94%**	35.37%
6-mer		-		min. free energy	43.54%	**44.22%**	42.86%
4-mer		-		min., avg. free energy	37.41%	**42.86%**	40.82%
5-mer		-		min., avg. free energy	**49.66%**	48.3%	42.86%
6-mer		-		min., avg. free energy	40.14%	**46.26%**	44.22%
4-mer		-		min., avg., max. free energy	**44.9%**	37.41%	35.37%
5-mer		-		min., avg., max. free energy	**42.86%**	36.73%	36.05%
6-mer		-		min., avg., max. free energy	**49.66%**	42.18%	27.89%
-	6	2	1	min. free energy	**39.46%**	35.37%	30.61%
-	7	1	1	min. free energy	36.73%	36.73%	**37.41%**
-	7	2	1	min. free energy	46.94%	**51.02%**	36.73%
-	9	1	1	min. free energy	**38.1%**	34.69%	33.33%
-	9	2	1	min. free energy	**51.02%**	48.98%	37.41%
-	11	1	1	min. free energy	**34.69%**	**34.69%**	25.85%
-	6	2	1	min., avg. free energy	**36.05%**	33.33%	31.97%
-	7	1	1	min., avg. free energy	32.65%	31.97%	**34.01%**
-	7	2	1	min., avg. free energy	38.78%	**44.9%**	**44.9%**
-	9	1	1	min., avg. free energy	**36.73%**	34.01%	34.69%
-	9	2	1	min., avg. free energy	**42.86%**	37.41%	36.73%
-	11	1	1	min., avg. free energy	**34.69%**	31.97%	33.33%
-	6	2	1	min., avg., max. free energy	27.89%	34.01%	**43.54%**
-	7	1	1	min., avg., max. free energy	36.05%	29.93%	**43.54%**
-	7	2	1	min., avg., max. free energy	**40.82%**	38.1%	35.37%
-	9	1	1	min., avg., max. free energy	38.78%	38.78%	**42.18%**
-	9	2	1	min., avg., max. free energy	**44.9%**	43.54%	39.46%
-	11	1	1	min., avg., max. free energy	**43.54%**	38.1%	35.37%
4-mer	6	2	1	min. free energy	**46.94%**	45.58%	40.82%
4-mer	7	1	1	min. free energy	35.37%	**41.5%**	29.93%
5-mer	7	2	1	min. free energy	**48.98%**	46.94%	45.58%
5-mer	9	1	1	min. free energy	**54.42%**	48.98%	37.41%
6-mer	9	2	1	min. free energy	36.73%	42.86%	**44.22%**
6-mer	11	1	1	min. free energy	**53.74%**	42.86%	36.05%
4-mer	6	2	1	min., avg. free energy	37.41%	**43.54%**	38.78%
4-mer	7	1	1	min., avg. free energy	**47.62%**	44.22%	44.9%
5-mer	7	2	1	min., avg. free energy	44.9%	**47.62%**	37.41%
5-mer	9	1	1	min., avg. free energy	40.82%	40.14%	**46.26%**
6-mer	9	2	1	min., avg. free energy	**53.06%**	32.65%	35.37%
6-mer	11	1	1	min., avg. free energy	**40.14%**	**40.14%**	36.05%
4-mer	6	2	1	min., avg., max. free energy	**46.94%**	45.58%	42.86%
4-mer	7	1	1	min., avg., max. free energy	39.46%	**45.58%**	42.18%
5-mer	7	2	1	min., avg., max. free energy	44.22%	**50.34%**	47.62%
5-mer	9	1	1	min., avg., max. free energy	**48.98%**	48.3%	42.86%
6-mer	9	2	1	min., avg., max. free energy	**45.58%**	42.18%	32.65%
6-mer	11	1	1	min., avg., max. free energy	**50.34%**	46.26%	40.14%

5.2.3 Comparison of the Different Secondary Structure Fingerprint Types

Overall, similar to our findings with the other dataset, when a single type of feature is used on its own (i.e. either a k-mer, a skip-mer, or a set of the secondary structure fingerprints), sequence-based features overall yielded superior classification performance [135]. The skip-mer with a length of 9 that alternatingly matches 2 consecutive nucleotides then skips 1 nucleotide yielded the highest accuracy of 84.92%, highest recall of 84.90%, and highest precision of 87.40% (using the 3-layered deep learning model), compared to using other standalone, non-combined features [135]. Meanwhile, the highest accuracy, precision, and recall achieved by the secondary structure fingerprints is only 59.42% (with the 5-layered model), 77.10% (with the 4-layered model), and 59.50% (with the 5-layered model) respectively, using the fingerprints that comprise of minimum, average, and maximum free energy values of matches combined together [135].

This trend in performance can also be observed when the reduced datasets are used for training and evaluation – sequence-based features performed better than the secondary structure fingerprints when the different features are used separately on their own. We find this rather surprising as we had expected and hypothesized that structural features would be advantageous as sequence similarity gets lower. With the dataset containing only sequences with up to 80% in similarity, the best performing accuracy between the 3 different neural network depths using each of the sequence-based feature sets ranged from 36.73% (match-1-skip-1 skip-mer of length 7, using the 4-layered deep neural network) to 54.42% (match-2-skip-1 skip-mer of length 7, using the 3-layered network architecture), while the secondary structure fingerprints counterparts ranged from 31.29% (the fingerprints that were built using only the minimum free energy values, used with the 5-layered model) to only 36.05% (fingerprints with the minimum, average, and maximum free energy values concantenated, using the 4-layered model) [135]. Similarly, when it comes to the other reduced dataset containing sequences of up to 90% in similarity, among the standalone sequence-based features, 6-mer with the 3-layered model achieved the highest accuracy of 80.51%; while the poorest performing feature is the match-1-skip-1 skip-mer of length 7, which reached a maximum accuracy of 52.08% using the 5-layered model configuration [135]. Meanwhile, the best performing standalone secondary structure fingerprints with the same reduced dataset was the fingerprints that combine minimum and average free energy values of matches, which only reached a maximum accuracy of 59.11% using the 5-layered model; while the poorest performing counterpart was the fingerprints that only use the minimum free energy values, which reached a low evaluation accuracy of 39.94% with the 5-layer model as well [135].

5.2.4 Combining Free Energy Values to Produce the Secondary Structure Fingerprints

Our original secondary structure fingerprints based on the curated common known RNA structural motifs only takes the match with the minimum free energy (i.e. most stable match in terms of thermodynamic stability) in case there are multiple secondary structure matches of a descriptor representing one of the curated secondary structure motifs [134]. However, for the RNA virus dataset, we also extracted the average free energy value and the match with the maximum free energy (i.e. least stable or poorest secondary structure match) when there are multiple matches of a descriptor in a sequence [135]. To determine how the extra free energy values contribute to or affect the overall classification performance, we used 3 different combinations of the values in the study: using the minimum free energy value only (same as our original common structural motifs based secondary structure fingerprints [134]); combining/concatenating minimum and average free energy values; and combining/concatenating the minimum, average, and maximum free energy values of the matches [135]. Combining/concatenating refers to including and using them in the feature vector – thus, if a secondary structure fingerprint of a sequence that only uses the minimum free energy value consists of 44 values (corresponding to the rescaled free energy values of the 44 curated secondary structure motifs), a counterpart that includes the minimum and average free energy values would consist of 88 values (44 rescaled minimum free energy values, and 44 rescaled average free energy values); while a fingerprint representing a sequence that uses the minimum, average, and maximum free energy values would consist of 132 values (from 44 minimum free energy values, 44 average values, and 44 maximum values) [135].

The results indicate that the additional free energy values are relevant to the host species identification, as the classification performance increased in all but one case when the additional values are included. For instance, with the primary dataset, the highest accuracy of the fingerprints that only use the minimum free energy is 36.75% (with the 4-layered model), while the accuracy of the fingerprints with the minimum and average free energy values is 52.60% (with the 5-layered model) [135]. The performance improved further when all of the minimum, average, and maximum free energy values are combined and used together, reaching a maximum 10-fold validated overall accuracy of 59.42% (with the 5-layers model) [135]. This increase can also be observed with the 10-fold validated precision: using only minimum free energy values yielded a highest precision of 69.70%, while using both minimum and average free energy values yielded 75.20% in precision, and using all three free energy values achieved a precision of 77.10% – all of which were obtained with the 4-layered deep neural network configuration. Meanwhile, the 10-fold validated recall increased from 36.60% (using only minimum free energy values, with the 4-layered deep learning model), to 52.60% (using minimum and average free energy values, with the 5-layered model), and further to 59.50% when all of three free energy values are used (with the 5-layered model).

There is a similar increase in accuracy when the reduced dataset with up to 80% sequence

similarity: from 31.29% (using minimum free energy only, and the 5-layered model), to 32.65% (minimum and average free energy, with the 5-layered model), and finally, 36.05% (minimum, average, and maximum free energy, with the 4-layered model). However, the case is different with the other reduced dataset which consists of sequences that are only up to 90% similar: using minimum free energy only yielded the lowest accuracy among all three of 39.94%, and using both minimum and average free energy values of matches yielded an improved accuracy of 59.11%; however, using all of the minimum, average, and maximum free energy values degraded the accuracy to 48.88% instead of improving it like in all of the other cases. This special case may or may not still be the same if 10-fold cross validation is performed with the reduced dataset, as it differed from the rest of the cases.

To summarize our findings, using the additional free energy values to form the secondary structure fingerprints resulted in an increase of classification performance in all except one of the cases. The additional improvements may be due to how RNA sequences themselves on their own may not fold into stable secondary structures without additional "folding kinetics" [58, 97]; and as a result, using the information from the non-optimal matches (i.e. represented by average and maximum free energy values of matches) retains representations of the secondary structures that could only be formed by the RNA sequences with the additional kinetics (which would otherwise be discarded if only the minimum free energy values are used, as the sequences themselves cannot fold into those structures optimally).

5.2.5 Combining Different Feature Types

Using more than one type of features does not always result in improved classification performance in this study. For instance, using 5-mer representation alone to identify the susceptible host species of a viral sequence resulted in a highest 10-fold validated overall accuracy of 72.52%, whereas adding match-1-skip-1 skip-mer of length 9 to be used together with the 5-mer as features degraded the highest overall accuracy to 69.97% [135].

Similarly, although we used feature sets with up to 3 different types of features; with the primary dataset, the best performing feature sets consists only of 2 types. When it comes to accuracy, a combination of the 6-mer and the skip-mer of length 9 that alternatingly matches 2 consecutive nucleotides and skips 1 nucleotides after achieved the highest 10-fold overall accuracy of 86.89% (using the 3-layered deep learning model) [135]. Meanwhile, in terms of precision, 6-mer combined with the match-1-skip-1 skip-mer of length 11 yielded the highest 10-fold precision of 88.6% (again, using the 3-layered model). Finally, when it comes to recall, the highest 10-fold overall recall of 87.0% was achieved by using 6-mer together with the 9 nucleotides long (including the skipped nucleotides) match-2-skip-1 skip-mer with the 3-layered deep learning configuration as well. None of the sets with 3 feature types combined exceed these 10-fold validated accuracy, precision, and recall [135].

5.2.6 Best Performing Feature Set is Different for the Reduced Datasets

We found that the feature set that performed best with the non-reduced primary dataset was outperformed by other sets when the reduced datasets are used for training and evaluation. For instance, with the reduced dataset of up to 80% sequence similarity, a combination of 5-mer and skip-mer of length 7 that matches 2 nucleotides and skips 1 nucleotide alternatingly, yielded the highest evaluation accuracy of 60.54% (using the model configuration with 3 layers) [135]. A different feature set achieved the best classification performance when it comes to the other reduced dataset of up to 90% sequence similarity – using 6-mer on its own yielded an accuracy of 80.51% (using the 3-layers deep learning model configuration as well) [135].

One major difference between the 3 different RNA virus datasets used in the study is their completeness. As the best performing feature set differed between the 3 datasets, including between the 2 different reduced datasets despite using the same set of host species, it is entirely possible that the best performing set of feature will also differ when our proposed approach is used with a more complete dataset in the future [135].

Chapter 6

Limitation

6.1 Limited Sequence Length

The RNA sequences in our studies were limited in length, as we found the RNAMotif [96] tool incorporated in our pipeline require large memory, and therefore significant amount of computational resources, to process longer sequences [134].

Based on our attempts in processing such longer sequences with RNAMotif, we observed that the memory usage of the program will increase until all of the available memory is used up, followed by a crash as no more additional free memory is available for it to allocate.

We found that this is especially true with our RAG-based secondary structure fingerprints, possibly because pseudoknots are taken into account. As a result, our study involving classification of non-coding RNA from the Rfam database [76, 77] only includes RNA sequences that are 148 nucleotides long or less [134]. Consequently, RNA classes that consist exclusively or mostly of longer sequences were excluded – one of such excluded classes was Intron Group I, despite the inclusion of Intron Group II in the study [134].

The maximum sequence length was significantly higher for our study involving viral RNA sequences – the longest sequence is 31,473 in length [135]. However, none of the secondary structure fingerprints take pseudoknots into account. In addition, only secondary structure fingerprints built without G·U wobble pairs were used, as we found that allowing the wobble pairs caused RNAMotif [96] to require larger memory for the longer sequences. In other words, if the wobble pairs are allowed, the maximum length of a sequence that can be processed with reasonable computational resources would be reduced, and as a result, the total entries in the dataset would also be limited.

In addition to the reduced included data entries and RNA classes, this current limitation also made it difficult to perform direct comparisons between our approach and similar previous

studies which involve longer sequences in their datasets, without access to exceptionally large computational resources.

6.2 No Distinction between Local and Global Matches by the Secondary Structure Fingerprints

One of the steps to build our secondary structure fingerprints is finding secondary structure matches of each descriptor in the set (representing the RAG graphs [44,50,68,101], or our curated common known RNA secondary structure motifs) that can be formed by the sequence [134]. The match can either be local, which means that the corresponding secondary structure can be formed by just a part of the sequence; or global, which means that the secondary structure would be formed by the entire sequence. However, the secondary structure fingerprints currently do not make the distinction between local or global matches. In other words, information on the aforementioned types of match is not retained in the fingerprints in any way.

As a result, RNAs that share "local substructures" with one or more different classes of RNAs, despite not sharing global structures, such as the HACA-box class of non-coding RNA [46], may be misclassified often when only the secondary structure fingerprints are used as deep learning features.

6.3 Use of RNAMotif Default Minimum Length for the RAG-based Secondary Structure Fingerprints

Our descriptors used to build the RAG-based secondary structure fingerprints initially specify a minimum consecutive paired nucleotides of 2 for a stem to be considered (which would correspond to an edge in the "dual graphs" RAG representation), a minimum number of consecutive nucleotides that was also specified by the authors of "RNA-As-Graphs" [44, 50]. However, this resulted in significant slowdown and increased memory requirement of RNAMotif [96] – we have provided up to 64 GB of RAM to process a single sequence with a single descriptor. As a result, we resorted to using the default minimum consecutive paired nucleotides of 3 instead of specifying the initial minimum of 2.

6.4 Complete Viral Sequence Requirement

The approach proposed in our study that involves predicting susceptible host species given a sequence of RNA virus has been developed and tested only with complete viral sequences, although

most of viral sequences from the NCBI Virus are indicated as "partial" sequences [18, 135]. This results in not only limited entries that could be used by our approach, but also the limited numbers of both viral species and susceptible host species that could be included, compared to if the approach were to work with both partial and complete viral sequences.

Chapter 7

Future Work

7.1 Improved Secondary Structure Fingerprints

7.1.1 Produce and Include Different Score Variations to Build Better Performing Fingerprints

Our findings indicate that producing different variations of scores and combining them, and combining different types of secondary structure fingerprints, produced better performing secondary structure fingerprints overall for classification purposes. For example, as previously discussed, in our study that uses secondary structure fingerprints to identify the type of an RNA given its sequence from the Rfam database [76, 77], combining the different scores from the 2 variations of the curated common structural motifs based fingerprints, one variant that allows wobble pairs and another that does not allow wobble pairs (i.e. concatenating two sets of 44 scores from the different variations of the fingerprints, to form a set of 88 scores which is then used as deep learning features), resulted in a higher accuracy than when each variation is used on their own as features [134]. Similarly, further increase of the 10-fold validated accuracy can be observed when this combination consisting of the wobble and non-wobble pair variants of the fingerprints, is further combined with the scores from the RAG-based fingerprints (forming fingerprints consisting of $88 + 47$ for each sequence, the latter being the number of scores in the RAG-based fingerprints per sequence) [134]. The same trend also applies to the precision and recall.

In another study of ours involving prediction of susceptible host species from a sequence of RNA virus; we found that in all cases, curated common motifs based secondary structure fingerprints that consist of both rescaled minimum and average free energy values of motif matches performed better compared to the fingerprints with only rescaled minimum free energy values [135].

In almost all cases, adding the rescaled maximum free energy values further improved the classification performance as well [135].

To summarize; based on our findings, producing different variations of scores or fingerprints, then combining them such that they are used together as deep learning features, will likely yield improved secondary structure fingerprints in terms of their resulting classification performance.

7.1.2 Include Information on Local vs. Global Matches

In order for the secondary structure fingerprints to be able to distinguish between different classes in which parts of the RNAs may share the same or similar secondary structures, but the RNA sequences in their entirety do not share the same secondary structure (for example, the HACA-box and CD-box [46]), the fingerprints need to contain information on whether the secondary structure matches are local (produced by only parts of the sequence) or global (produced by the entire sequence).

One possible approach to achieve this is by using two sets of scores instead of one for a single type of secondary structure fingerprints: one set is derived only from local matches with the descriptors, while the other set is derived only from global matches. In other words, local secondary structure matches will only affect one of the sets, while global structural matches will only affect the other set. Thus, in case of different RNA classes with differing global secondary structures but similar local secondary structures, the set of scores derived from only global matches would be different; and these differences in the resulting fingerprints could then indicate the different RNA classes.

7.1.3 Include Positional Information of Secondary Structure Matches

In addition to including information on whether a secondary structure match is local or global in the fingerprints, positional information of structural matches could also be encoded and included in them. Presently, there is no such information in the fingerprints – the fingerprints currently do not differentiate whether a secondary structure match is at the beginning, middle, or end of a RNA strand. A future study could develop secondary structure fingerprints with such positional information included in them, and assess if the positional information improved the resulting classification performance.

Improvements have indeed been shown by results of our study that extends the work covered in this thesis [133]. This extended study separates the match scores produced by, and in turn, differentiates between global and local structural matches [133]. The overall results validated our hypothesis that incorporating positional information in the secondary structure fingerprints

would improve the classification performance. In fact, the best classification performance in the study is achieved when the secondary structure fingerprints are used along with k-mer [133].

7.2 More Efficient Secondary Structure Fingerprints Pipeline

7.2.1 Use Smaller but More Descriptors

One way to reduce the memory requirements of RNAMotif [96] used in our secondary structure fingerprints pipeline is by using smaller or shorter descriptors – and to make up the shorter lengths, additional descriptors could be written and used [135]. A future study that incorporates the same pipeline using RNAMotif [96] to produce secondary structure fingerprints could focus on creating such smaller but more numerous descriptors, including how to split the original larger descriptor into smaller ones without breaking the connections – i.e. the smaller descriptors that were broken down from a larger descriptor should still be connected as opposed to becoming completely separate descriptors. For example, given a group of small descriptors that were broken down from a single larger descriptor, one approach to achieve the simulated connectivity is by ignoring any secondary structure matches in one small descriptor if there is no secondary structure match with the other small descriptors in the group (as the large original descriptor would not produce a match either in this case).

7.2.2 RNAMotif Substitution

RNAMotif plays a key role in the pipeline to create secondary structure fingerprints from RNA sequences. However, as already mentioned, this software requires significant computational resources. In order to circumvent this limitation, a new tool could be built to match RNA secondary structure motifs.

To ensure computational efficiency, the tool could leverage approaches used by the Seed program [8]. Seed is based on an efficient data structure, called suffix array, that allows simultaneous search of all starting positions of the input sequence for a given motif. This makes the search time proportional to the size of the motif, and not the size of the input sequence. The preprocessing time needed to build the required indices is linear with respect to the size of the input sequence. Likewise, the memory requirement for the indices also grows linearly with the size of the input sequence. Consequently, it is reasonable to expect that such pattern matching tool would be able to process the longer sequences that were excluded in our studies. Furthermore, a dedicated tool could be optimized for our needs. For example, certain parts of the search space could be pruned if deemed unnecessary.

Herein, the structural features were either extracted from RAG or manually curated. Another interesting avenue to pursue is developing a method that automatically learns and selects a set of structural features that would yield the highest classification performance. Thus, structural features that are not informational for a particular dataset would not be used. Again, the techniques used by Seed could be leveraged.

7.3 Classification with Partial Sequences

For some sources of data, including the NCBI Virus [18] we used for our study [135], partial sequences are more abundant than complete sequences. Extending our approach such that the classifier works with partial sequences would be useful for datasets and cases where the newly discovered sequences (e.g. newer RNA viruses) are only partially available [135].

Chapter 8

Conclusion

We proposed a pipeline to produce secondary structure fingerprints from RNA sequences, which represent the possible secondary structures that can be formed by the RNA sequence, and can be used as input features in order to identify or classify the sequences. This approach is based on the facts that it is possible for a single strand of RNA to form multiple secondary structures [140]; and that RNA functions may depend on the secondary structures [140], thus secondary structure information can be used to infer the function and consequently the identity of the RNA.

Our proposed pipeline was used to produce different sets of secondary structure fingerprints: a set based on "RNA-As-Graphs" [44, 50, 68, 101], which takes pseudoknots into account; another based on our curated common known RNA secondary structure motifs [134], which takes free energy into account, as the scores comprising the features are derived from free energy values of matches with the curated motifs.

We then used each type of the produced secondary structure fingerprints as deep learning features, with different datasets to solve different RNA classification problems, alone and in conjunction with the other types of secondary structure fingerprints and/or sequence-based features (continuous k-mers [134, 135], and non-continuous k-mers/"skip-mers" [31, 135]); in order to assess how the classification performance is affected. The first dataset contains non-coding RNAs from Rfam database version 14.1 [76, 77, 134], and the RNAs are to be classified using the features with deep learning into their different RNA classes [134]. The second dataset contains sequences of RNA virus from NCBI Virus [18], and our study with this dataset involved identifying host species that are susceptible given the viral sequences [135].

Our findings indicate that the sequence-based features overall outperformed our current secondary structure fingerprints. Combining the secondary structure fingerprints with poorer performing sequence-based features (for example, 2-mer and 3-mer, but not 4-mer in our study involving classification of non-coding RNAs [134]) resulted in improvements in the final classi-

fication performance. However, there is no improvements but slight degradation in classification performance, when sequence-based features that already perform well (e.g. 4-mer in our non-coding RNA study [134]) are used together with the secondary structure fingerprints [134]. Similarly, adding secondary structure fingerprints to the best performing combination of k-mer and skip-mer in our RNA virus host prediction study, such that they are used together as deep learning features, resulted in a poorer classification performance compared to when the secondary structure fingerprints are not used [135]. Since using the secondary structure fingerprints as additional deep learning features together with sequence-based features that already perform well did not result in further improvements, our initial hypothesis is invalidated.

Despite of these findings regarding the secondary structure fingerprints, we found that our deep learning approach performed well with the sequence-based features for both datasets and classification problems – a maximum 10-fold validated classification accuracy of 86.92% was achieved using 4-mer in our non-coding RNA classification study [134], while 86.89% was achieved by a feature set that combines 6-mer and match-2-skip-1 skip-mer with a length of 9 in our other study involving host susceptibility to RNA virus [135].

In addition, we also found that producing and combining different variants of secondary structure fingerprints resulted in better performing fingerprints. For example, combining secondary structure fingerprints based on our curated common structural motifs that both allow and disallow wobble pairs together resulted in higher classification accuracy, compared to when only one of the 2 variants is used separately [134]. This also applies all but one case in our other study [135]. Thus, a future study should produce different variants of the fingerprints with different rules (e.g. wobble pairs allowed and disallowed), and combine/concatenate them such that they are used together as deep learning features, in order to have a better performing secondary structure fingerprints in terms of classification performance.

Furthermore, our current secondary structure fingerprints do not differentiate between local and global structures. For instance, if there is a specific secondary structure that can formed by only a part of specific sequence, while another sequence can form the same secondary structure but the whole sequence is involved, the secondary structure fingerprints for the different sequences will not differentiate between the difference in scope. As a result, different RNAs that share similar local structures but do not share similar global structures may not be well differentiated by the current secondary structure fingerprints [46]. Future development of the secondary structure fingerprints could split the different scopes of matches into different scores, such that the difference can then be inferred by machine/deep learning. Additionally, the current fingerprints also do not include any positional information on the structural matches. Such information can be encoded in a future version of the fingerprints, which may be useful for the neural network in order to classify/identify the RNAs correctly.

Finally, our current pipeline involving RNAMotif [96] require large amounts of memory and computational resources to process longer sequences. Future studies that would like to adopt our proposed pipeline could use smaller but more RNAMotif descriptors [96] in order to reduce the

memory requirements. Another alternative is to create a custom and efficient secondary structure matching solution specific to the pipeline, to be used in place of RNAMotif.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Paul F Agris. Bringing order to translation: the contributions of transfer RNA anticodon-domain modifications. *EMBO reports*, 9(7):629–635, 2008.

[3] Nathan A Ahlgren, Jie Ren, Yang Young Lu, Jed A Fuhrman, and Fengzhu Sun. Alignment-free d_2^* oligonucleotide frequency dissimilarity measure improves prediction of hosts from metagenomically-derived viral sequences. *Nucleic Acids Research*, 45(1):39–53, 11 2016.

[4] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, Antalya, August 2017. IEEE.

[5] Noorul Amin, Annette McGrath, and Yi-Ping Phoebe Chen. Evaluation of deep learning in non-coding RNA classification. *Nature Machine Intelligence*, 1(5):246–256, May 2019.

[6] Eleni Anastasiadou, Leni S. Jacob, and Frank J. Slack. Non-coding RNA networks in cancer. *Nature Reviews Cancer*, 18(1):5–18, January 2018.

[7] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology. *Molecular Systems Biology*, 12(7):878, July 2016.

[8] Mohammad Anwar, Truong Nguyen, and Marcel Turcotte. Identification of consensus RNA secondary structures using suffix arrays. *BMC Bioinformatics*, 7(1):244, 2006.

[9] Antonio Arauzo-Azofra, Jose Manuel Benitez, and Juan Luis Castro. Consistency measures for feature selection. *Journal of Intelligent Information Systems*, 30(3):273–292, June 2008.

[10] Junghwan Baek, Byunghan Lee, Sunyoung Kwon, and Sungroh Yoon. LncRNAnet: long non-coding RNA identification using deep learning. *Bioinformatics*, 34(22):3889–3897, November 2018.

[11] M Z Barciszewska, M Szymański, V A Erdmann, and J Barciszewski. Structure and functions of 5S rRNA. *Acta Biochimica Polonica*, 48(1):191–198, March 2001.

[12] Daniel Berrar. Cross-validation. *Encyclopedia of Bioinformatics and Computational Biology*, 1:542–545, 2019.

[13] Karel Břinda, Maciej Sykulski, and Gregory Kucherov. Spaced seeds improve k-mer-based metagenomic classification. *Bioinformatics*, 31(22):3584–3592, November 2015.

[14] Eckart Bindewald, Robert Hayes, Yaroslava G. Yingling, Wojciech Kasprzak, and Bruce A. Shapiro. RNAJunction: a database of RNA junctions and kissing loops for three-dimensional structural analysis and nanodesign. *Nucleic Acids Research*, 36(suppl_1):D392–D397, January 2008.

[15] Marcus Boden, Martin Schöneich, Sebastian Horwege, Sebastian Lindner, Chris Leimeister, and Burkhard Morgenstern. Alignment-free sequence comparison with spaced k-mers. In Tim Beißbarth, Martin Kollmar, Andreas Leha, Burkhard Morgenstern, Anne-Kathrin Schultz, Stephan Waack, and Edgar Wingender, editors, *German Conference on Bioinformatics 2013*, volume 34 of *OpenAccess Series in Informatics (OASIcs)*, pages 24–34, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[16] Christian Borgelt, Thorsten Meinl, and Michael Berthold. MoSS: a program for molecular substructure mining. In *Proceedings of the 1st international workshop on open source data mining frequent pattern mining implementations - OSDM '05*, pages 6–15, Chicago, Illinois, 2005. ACM Press.

[17] Ian Brierley, Simon Pennell, and Robert J. C. Gilbert. Viral RNA pseudoknots: versatile motifs in gene expression and replication. *Nature Reviews Microbiology*, 5(8):598–610, August 2007.

[18] J. Rodney Brister, Danso Ako-adjei, Yiming Bao, and Olga Blinkova. NCBI viral genomes resource. *Nucleic Acids Research*, 43(D1):D571–D577, January 2015.

[19] Yogev Brown, Mira Abraham, Sivan Pearl, Majdi M. Kabaha, Elhanan Elboher, and Yehuda Tzfati. A critical three-way junction is conserved in budding yeast and vertebrate telomerase RNAs. *Nucleic Acids Research*, 35(18):6280–6289, September 2007.

[20] Christopher JC Burges, Bernhard Scholkopf, and Alexander J Smola. *Advances in kernel methods: support vector learning*. MIT press Cambridge, MA, USA:, 1999.

[21] Donald Burke. Recombination in HIV: An important viral evolutionary strategy. *Emerging Infectious Diseases*, 3(3):253–259, September 1997.

[22] Sara A. Byron, Kendall R. Van Keuren-Jensen, David M. Engelthaler, John D. Carpten, and David W. Craig. Translating RNA sequencing into clinical diagnostics: opportunities and challenges. *Nature Reviews Genetics*, 17(5):257–271, May 2016.

[23] Zhen Cao and Shihua Zhang. Probe efficient feature representation of gapped k-mer frequency vectors from sequences using deep neural networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(2):657–667, March 2020.

[24] Thomas R. Cech. The efficiency and versatility of catalytic RNA: implications for an RNA world. *Gene*, 135(1-2):33–36, December 1993.

[25] Mohamed Chaabane, Robert M Williams, Austin T Stephens, and Juw Won Park. circDeep: deep learning approach for circular RNA classification from other long non-coding RNA. *Bioinformatics*, 36(1):73–80, January 2020.

[26] Jia Chan and Yvonne Tay. Noncoding RNA: RNA regulatory networks in cancer. *International Journal of Molecular Sciences*, 19(5):1310, Apr 2018.

[27] Jia Cheng, Jun-Ming Guo, Bing-Xiu Xiao, Ying Miao, Zhen Jiang, Hui Zhou, and Qing-Ning Li. piRNA, the new non-coding RNA, is aberrantly expressed in human cancer cells. *Clinica Chimica Acta*, 412(17-18):1621–1625, August 2011.

[28] Liam Childs, Zoran Nikoloski, Patrick May, and Dirk Walther. Identification and classification of ncRNA molecules using graph properties. *Nucleic Acids Research*, 37(9):e66–e66, May 2009.

[29] Dami Choi, Christopher J. Shallue, Zachary Nado, Jaehoon Lee, Chris J. Maddison, and George E. Dahl. On empirical comparisons of optimizers for deep learning. *arXiv:1910.05446 [cs, stat]*, June 2020. arXiv: 1910.05446.

[30] François Chollet et al. Keras. https://keras.io, 2015.

[31] Bernardo J. Clavijo, Gonzalo Garcia Accinelli, Luis Yanes, Katie Barr, and Jonathan Wright. Skip-mers: increasing entropy and sensitivity to detect conserved genic regions with simple cyclic q-grams. preprint, Bioinformatics, August 2017.

[32] P. Clote. Structural RNA has lower folding energy than random RNA of the same dinucleotide frequency. *RNA*, 11(5):578–591, May 2005.

[33] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. L2 regularization for learning kernels. *arXiv:1205.2653 [cs, stat]*, May 2012. arXiv: 1205.2653.

[34] Valerio Costa, Claudia Angelini, Italia De Feis, and Alfredo Ciccodicola. Uncovering the complexity of transcriptomes with RNA-Seq. *Journal of Biomedicine and Biotechnology*, 2010:1–19, 2010.

[35] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In Matthieu Cord and Pádraig Cunningham, editors, *Machine Learning Techniques for Multimedia*, pages 21–49. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. Series Title: Cognitive Technologies.

[36] Frank E. Curtis and Katya Scheinberg. Optimization methods for supervised machine learning: From linear models to deep learning. In Rajan Batta, Jiming Peng, J. Cole Smith, and Harvey J. Greenberg, editors, *The Operations Research Revolution*, pages 89–113. INFORMS, September 2017.

[37] Padideh Danaee, Mason Rouches, Michelle Wiley, Dezhong Deng, Liang Huang, and David Hendrix. bpRNA: large-scale automated annotation and analysis of RNA secondary structure. *Nucleic Acids Research*, 46(11):5381–5394, June 2018.

[38] Ayon Dey. Machine learning algorithms: a review. *International Journal of Computer Science and Information Technologies*, 7(3):1174–1179, 2016.

[39] Kishore J Doshi, Jamie J Cannone, Christian W Cobaugh, and Robin R Gutell. Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for RNA secondary structure prediction. *BMC Bioinformatics*, 5:105, Aug 2004.

[40] Veronika B. Dubinkina, Dmitry S. Ischenko, Vladimir I. Ulyantsev, Alexander V. Tyakht, and Dmitry G. Alexeev. Assessment of k-mer spectrum applicability for metagenomic dissimilarity analysis. *BMC Bioinformatics*, 17(1):38, December 2016.

[41] Rob A Dunne and Norm A Campbell. On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. In *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne*, volume 181, page 185. Citeseer, 1997.

[42] Konstantin Eckle and Johannes Schmidt-Hieber. A comparison of deep networks with ReLU activation function and linear spline-type methods. *Neural Networks*, 110:232–242, February 2019.

[43] Jörg Fallmann, Sebastian Will, Jan Engelhardt, Björn Grüning, Rolf Backofen, and Peter F. Stadler. Recent advances in RNA folding. *Journal of Biotechnology*, 261:97–104, November 2017.

[44] Daniela Fera, Namhee Kim, Nahum Shiffeldrim, Julie Zorn, Uri Laserson, Hin Gan, and Tamar Schlick. RAG: RNA-As-Graphs web resource. *BMC Bioinformatics*, 5(1):88, 2004.

[45] Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated Machine Learning*, pages 3–33. Springer, Cham, 2019.

[46] Antonino Fiannaca, Massimo La Rosa, Laura La Paglia, Riccardo Rizzo, and Alfonso Urso. nRC: non-coding RNA classifier based on structural features. *BioData Mining*, 10(1):27, December 2017.

[47] Alexandra Forsbach, Jean-Guy Nemorin, Carmen Montino, Christian Müller, Ulrike Samulowitz, Alain P. Vicari, Marion Jurk, George K. Mutwiri, Arthur M. Krieg, Grayson B. Lipford, and Jörg Vollmer. Identification of RNA Sequence Motifs Stimulating Sequence-Specific TLR8-Dependent Immune Responses. *The Journal of Immunology*, 180(6):3729–3738, March 2008.

[48] Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, December 2012.

[49] Clovis Galiez, Matthias Siebert, François Enault, Jonathan Vincent, and Johannes Söding. WIsH: who is the host? predicting prokaryotic hosts from metagenomic phage contigs. *Bioinformatics*, 33(19):3113–3114, 07 2017.

[50] H. H. Gan, D. Fera, J. Zorn, N. Shiffeldrim, M. Tang, U. Laserson, N. Kim, and T. Schlick. RAG: RNA-As-Graphs database–concepts, analysis, and features. *Bioinformatics*, 20(8):1285–1291, May 2004.

[51] Mahmoud Ghandi, Dongwon Lee, Morteza Mohammad-Noori, and Michael A. Beer. Enhanced regulatory sequence prediction using gapped k-mer features. *PLoS Computational Biology*, 10(7):e1003711, July 2014.

[52] David P Giedroc, Carla A Theimer, and Paul L Nixon. Structure, stability and function of RNA pseudoknots involved in stimulating ribosomal frameshifting. *Journal of Molecular Biology*, 298(2):167–185, April 2000.

[53] Antarip Halder, Dhruv Data, Preethi P. Seelam, Dhananjay Bhattacharyya, and Abhijit Mitra. Estimating strengths of individual hydrogen bonds in RNA base pairs: Toward a consensus between different computational approaches. *ACS Omega*, 4(4):7354–7368, April 2019.

[54] Nicholas B. Hammond, Blanton S. Tolbert, Ryszard Kierzek, Douglas H. Turner, and Scott D. Kennedy. RNA internal loops with tandem AG pairs: The structure of the 5'G AG U/3'U GA G loop can be dramatically different from others, including 5'A AG U/3'U GA A. *Biochemistry*, 49(27):5817–5827, July 2010.

[55] Peter Harrington. *Machine Learning in Action*. Manning Publications Co., USA, 2012.

[56] Masami Hasegawa and Takashi Miyata. On the antisymmetry of the amino acid code table. *Origins of Life*, 10(3):265–270, September 1980.

[57] William Grant Hatcher and Wei Yu. A survey of deep learning: Platforms, applications and emerging research trends. *IEEE Access*, 6:24411–24432, 2018.

[58] S. L. Heilman-Miller. Effect of transcription on folding of the Tetrahymena ribozyme. *RNA*, 9(6):722–733, June 2003.

[59] Daniel Herschlag. RNA chaperones and the RNA folding problem. *Journal of Biological Chemistry*, 270(36):20871–20874, September 1995.

[60] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[61] I. L. Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Research*, 31(13):3429–3431, July 2003.

[62] Ivo L Hofacker and Peter F Stadler. Automatic detection of conserved base pairing patterns in RNA virus genomes. *Computers & Chemistry*, 23(3-4):401–414, June 1999.

[63] Edward C. Holmes and Andrew Rambaut. Viral evolution and the emergence of SARS coronavirus. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 359(1447):1059–1065, July 2004.

[64] Hiroshi Hori and Syozo Osawa. Evolutionary change in 5S rRNA secondary structure and a phylogenic tree of 352 5S rRNA species. *Biosystems*, 19(3):163–172, January 1986.

[65] J. L. Hyde, C. L. Gardner, T. Kimura, J. P. White, G. Liu, D. W. Trobaugh, C. Huang, M. Tonelli, S. Paessler, K. Takeda, W. B. Klimstra, G. K. Amarasinghe, and M. S. Diamond. A Viral RNA Structural Element Alters Host Recognition of Nonself RNA. *Science*, 343(6172):783–787, February 2014.

[66] J. L. Hyde, C. L. Gardner, T. Kimura, J. P. White, G. Liu, D. W. Trobaugh, C. Huang, M. Tonelli, S. Paessler, K. Takeda, W. B. Klimstra, G. K. Amarasinghe, and M. S. Diamond. A viral RNA structural element alters host recognition of nonself RNA. *Science*, 343(6172):783–787, February 2014.

[67] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, July 2019.

[68] Joseph A Izzo, Namhee Kim, Shereef Elmetwaly, and Tamar Schlick. RAG: An update to the RNA-As-Graphs resource. *BMC Bioinformatics*, 12(1):219, 2011.

[69] Luc Jaeger, François Michel, and Eric Westhof. Involvement of a GNRA tetraloop in long-range RNA tertiary interactions. *Journal of Molecular Biology*, 236(5):1271–1276, March 1994.

[70] A. Jambhekar and J. L. DeRisi. Cis-acting determinants of asymmetric, cytoplasmic RNA transport. *RNA*, 13(5):625–642, May 2007.

[71] Eckhard Jankowsky and Michael E. Harris. Specificity and nonspecificity in RNA–protein interactions. *Nature Reviews Molecular Cell Biology*, 16(9):533–544, September 2015.

[72] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *arXiv:1702.05659 [cs]*, February 2017. arXiv: 1702.05659.

[73] Andrew D. Johnson. An extended IUPAC nomenclature code for polymorphic nucleic acids. *Bioinformatics*, 26(10):1386–1389, May 2010.

[74] T. H. Jukes. Relations between mutations and base sequences in the amino acid code. *Proceedings of the National Academy of Sciences*, 48(10):1809–1815, October 1962.

[75] Daniel Justus, John Brennan, Stephen Bonner, and Andrew Stephen McGough. Predicting the computational cost of deep learning models. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3873–3882, Seattle, WA, USA, December 2018. IEEE.

[76] Ioanna Kalvari, Joanna Argasinska, Natalia Quinones-Olvera, Eric P Nawrocki, Elena Rivas, Sean R Eddy, Alex Bateman, Robert D Finn, and Anton I Petrov. Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Research*, 46(D1):D335–D342, January 2018.

[77] Ioanna Kalvari, Eric P. Nawrocki, Joanna Argasinska, Natalia Quinones-Olvera, Robert D. Finn, Alex Bateman, and Anton I. Petrov. Non-coding RNA analysis using the Rfam database. *Current Protocols in Bioinformatics*, 62(1):e51, June 2018.

[78] Nils Kasties, Eric Jandciu, Alan Jones, Michael Wink, and Renate FitzRoy. *An introduction to molecular biotechnology*. John Wiley & Sons, 2006.

[79] Kevin S. Keating, Navtej Toor, and Anna Marie Pyle. The GANC tetraloop: a novel motif in the Group IIC intron structure. *Journal of Molecular Biology*, 383(3):475–481, November 2008.

[80] Tae Hoon Kim, Ziedulla K. Abdullaev, Andrew D. Smith, Keith A. Ching, Dmitri I. Loukinov, Roland D. Green, Michael Q. Zhang, Victor V. Lobanenkov, and Bing Ren. Analysis of the vertebrate insulator protein CTCF-binding sites in the human genome. *Cell*, 128(6):1231–1245, March 2007.

[81] Jessime M. Kirk, Susan O. Kim, Kaoru Inoue, Matthew J. Smola, David M. Lee, Megan D. Schertzer, Joshua S. Wooten, Allison R. Baker, Daniel Sprague, David W. Collins, Christopher R. Horning, Shuo Wang, Qidi Chen, Kevin M. Weeks, Peter J. Mucha, and J. Mauro Calabrese. Functional classification of long non-coding RNAs by k-mer content. *Nature Genetics*, 50(10):1474–1482, October 2018.

[82] Grzegorz Kondrak. N-gram similarity and distance. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Mariano Consens, and Gonzalo Navarro, editors, *String Processing and Information Retrieval*, volume 3772, pages 115–126. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. Series Title: Lecture Notes in Computer Science.

[83] Jessica Koplin, Yuguang Mu, Christian Richter, Harald Schwalbe, and Gerhard Stock. Structure and dynamics of an RNA tetraloop: A joint molecular dynamics and NMR study. *Structure*, 13(9):1255–1267, September 2005.

[84] Marcel Kucharík, Ivo L. Hofacker, Peter F. Stadler, and Jing Qin. Pseudoknots in RNA folding landscapes. *Bioinformatics*, page btv572, October 2015.

[85] Jens Kurreck. RNA interference: From basic research to therapeutic applications. *Angewandte Chemie International Edition*, 48(8):1378–1398, February 2009.

[86] Eric C. Lai. Micro RNAs are complementary to 3' UTR sequence motifs that mediate negative post-transcriptional regulation. *Nature Genetics*, 30(4):363–364, April 2002.

[87] Jung C. Lee, Robin R. Gutell, and Rick Russell. The UAA/GAN internal loop motif: A new RNA structural element that forms a cross-strand AAA stack and long-range tertiary interactions. *Journal of Molecular Biology*, 360(5):978–988, July 2006.

[88] S. Lemieux. RNA canonical and non-canonical base pairing types: a recognition method and complete repertoire. *Nucleic Acids Research*, 30(19):4250–4263, October 2002.

[89] Neocles B Leontis and Eric Westhof. Analysis of RNA motifs. *Current Opinion in Structural Biology*, 13(3):300–308, June 2003.

[90] N Leulliot. Unusual nucleotide conformations in GNRA and UNCG type tetraloop hairpins: evidence from Raman markers assignments. *Nucleic Acids Research*, 27(5):1398–1404, March 1999.

[91] W. Li and A. Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, July 2006.

[92] Yan Li, Qiupeng Zheng, Chunyang Bao, Shuyi Li, Weijie Guo, Jiang Zhao, Di Chen, Jianren Gu, Xianghuo He, and Shenglin Huang. Circular RNA is enriched and stable in exosomes: a promising biomarker for cancer diagnosis. *Cell Research*, 25(8):981–984, August 2015.

[93] Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. *arXiv:1910.07454 [cs, stat]*, November 2019. arXiv: 1910.07454.

[94] Bin Liu, Longyun Fang, Shanyi Wang, Xiaolong Wang, Hongtao Li, and Kuo-Chen Chou. Identification of microRNA precursor with the degenerate K-tuple or Kmer strategy. *Journal of Theoretical Biology*, 385:153–159, November 2015.

[95] Han Liu and Mihaela Cocea. Semi-random partitioning of data into training and test sets in granular computing context. *Granular Computing*, 2(4):357–386, December 2017.

[96] T. J. Macke. RNAMotif, an RNA secondary structure definition and search algorithm. *Nucleic Acids Research*, 29(22):4724–4735, November 2001.

[97] David H. Mathews. Revolutions in RNA secondary structure prediction. *Journal of Molecular Biology*, 359(3):526–532, June 2006.

[98] David H. Mathews, Jeffrey Sabina, Michael Zuker, and Douglas H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288(5):911–940, May 1999.

[99] David H Mathews and Douglas H Turner. Prediction of RNA secondary structure by free energy minimization. *Current Opinion in Structural Biology*, 16(3):270–278, June 2006.

[100] R. McDaniell, B. K. Lee, L. Song, Z. Liu, A. P. Boyle, M. R. Erdos, L. J. Scott, M. A. Morken, K. S. Kucera, A. Battenhouse, D. Keefe, F. S. Collins, H. F. Willard, J. D. Lieb, T. S. Furey, G. E. Crawford, V. R. Iyer, and E. Birney. Heritable individual-specific and allele-specific chromatin signatures in humans. *Science*, 328(5975):235–239, April 2010.

[101] Grace Meng, Marva Tariq, Swati Jain, Shereef Elmetwaly, and Tamar Schlick. RAG-Web: RNA structure prediction/design using RNA-As-Graphs. *Bioinformatics*, page btz611, August 2019.

[102] Florian Mock, Adrian Viehweger, Emanuel Barth, and Manja Marz. VIDHOP, viral host prediction with deep learning. *Bioinformatics*, 08 2020. btaa705.

[103] A. M. Molinaro, R. Simon, and R. M. Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, August 2005.

[104] Marco Molinaro and Ignacio Tinoco. Use of ultra stable UNCG tetraloop hairpins to fold RNA structures: thermodynamic and spectroscopic applications. *Nucleic Acids Research*, 23(15):3056–3063, 1995.

[105] David R. Morris and Adam P. Geballe. Upstream open reading frames as regulators of mRNA translation. *Molecular and Cellular Biology*, 20(23):8635–8642, December 2000.

[106] U. Nagaswamy. NCIR: a database of non-canonical interactions in known RNA structures. *Nucleic Acids Research*, 30(1):395–397, January 2002.

[107] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[108] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1, December 2015.

[109] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv:1811.03378 [cs]*, November 2018. arXiv: 1811.03378.

[110] Boel Nyström and Lennart Nilsson. Molecular dynamics study of intrinsic stability in six RNA terminal loop motifs. *Journal of Biomolecular Structure and Dynamics*, 24(6):525–535, June 2007.

[111] Xiaoyong Pan and Hong-Bin Shen. Learning distributed representations of RNA sequences and its application for predicting RNA-protein binding sites with a convolutional neural network. *Neurocomputing*, 305:51–58, August 2018.

[112] Xiaoyong Pan and Hong-Bin Shen. Predicting RNA–protein binding sites and motifs through combining local and global deep convolutional neural networks. *Bioinformatics*, 34(20):3427–3436, October 2018.

[113] Bharat Panwar, Amit Arora, and Gajendra PS Raghava. Prediction and classification of ncRNAs using structural information. *BMC Genomics*, 15(1):127, 2014.

[114] Alla Peselis and Alexander Serganov. Structure and function of pseudoknots involved in gene expression control: Structure and function of pseudoknots. *Wiley Interdisciplinary Reviews: RNA*, 5(6):803–822, November 2014.

[115] David M. W. Powers. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv:2010.16061 [cs, stat]*, October 2020. arXiv: 2010.16061.

[116] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *When Training and Test Sets Are Different: Characterizing Learning Transfer*, pages 3–28. MIT Press, 2009.

[117] Irfan A. Qureshi and Mark F. Mehler. The emerging role of epigenetics in stroke: II. RNA regulatory circuitry. *Archives of Neurology*, 67(12):1435–1441, 12 2010.

[118] Ramya Rangan, Ivan N. Zheludev, Rachel J. Hagey, Edward A. Pham, Hannah K. Wayment-Steele, Jeffrey S. Glenn, and Rhiju Das. RNA genome conservation and secondary structure in SARS-CoV-2 and SARS-related viruses: a first look. *RNA*, 26(8):937–959, August 2020.

[119] Jie Ren, Nathan A. Ahlgren, Yang Young Lu, Jed A. Fuhrman, and Fengzhu Sun. VirFinder: a novel k-mer based tool for identifying viral sequences from assembled metagenomic data. *Microbiome*, 5(1):69, December 2017.

[120] Peter W. Rose, Andreas Prlić, Ali Altunkaya, Chunxiao Bi, Anthony R. Bradley, Cole H. Christie, Luigi Di Costanzo, Jose M. Duarte, Shuchismita Dutta, Zukang Feng, Rachel Kramer Green, David S. Goodsell, Brian Hudson, Tara Kalro, Robert Lowe, Ezra Peisach, Christopher Randle, Alexander S. Rose, Chenghua Shao, Yi-Ping Tao, Yana Valasatava, Maria Voigt, John D. Westbrook, Jesse Woo, Huangwang Yang, Jasmine Y. Young, Christine Zardecki, Helen M. Berman, and Stephen K. Burley. The RCSB protein data bank: integrative view of protein, gene and 3D structural information. *Nucleic Acids Research*, 45(D1):D271–D281, 10 2016.

[121] R Saldanha, G Mohr, M Belfort, and A M Lambowitz. Group I and group II introns. *The FASEB Journal*, 7(1):15–24, January 1993.

[122] Kengo Sato, Yuki Kato, Michiaki Hamada, Tatsuya Akutsu, and Kiyoshi Asai. IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. *Bioinformatics*, 27(13):i85–i93, July 2011.

[123] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, January 2015.

[124] José A. Sáez, Julián Luengo, and Francisco Herrera. Evaluating the classifier behavior with noisy data considering performance and robustness: The Equalized Loss of Accuracy measure. *Neurocomputing*, 176:26–35, February 2016.

[125] Babak Shakibi. *Predicting parameters in deep learning*. PhD thesis, University of British Columbia, 2014.

[126] P. Simmonds. Pervasive RNA secondary structure in the genomes of SARS-CoV-2 and other coronaviruses. *mBio*, 11(6):e01661–20, /mbio/11/6/mBio.01661–20.atom, October 2020.

[127] P. Simmonds and D. B. Smith. Structural constraints on RNA virus evolution. *Journal of Virology*, 73(7):5787–5794, July 1999.

[128] Richard Simon. Resampling strategies for model assessment and selection. In *Fundamentals of data mining in genomics and proteomics*, pages 173–186. Springer, 2007.

[129] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, July 2009.

[130] Stephen Solis-Reyes, Mariano Avino, Art Poon, and Lila Kari. An open-source k-mer based machine learning tool for fast and accurate subtyping of HIV-1 genomes. *PLOS ONE*, 13(11):e0206409, November 2018.

[131] N. Spackova. Molecular dynamics simulations of sarcin-ricin rRNA motif. *Nucleic Acids Research*, 34(2):697–708, January 2006.

[132] David W Staple and Samuel E Butcher. Pseudoknots: RNA structures with diverse functions. *PLoS Biology*, 3(6):e213, June 2005.

[133] Kevin Sutanto and Marcel Turcotte. Assessing global-local secondary structure fingerprints to classify RNA sequences with deep learning. Submitted 2021-02-28.

[134] Kevin Sutanto and Marcel Turcotte. Assessing the use of secondary structure fingerprints and deep learning to classify RNA sequences. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 42–49, Seoul, Korea (South), December 2020. IEEE.

[135] Kevin Sutanto and Marcel Turcotte. Extracting and evaluating features from RNA virus sequences to predict host species susceptibility using deep learning. In *International Conference on Bioinformatics and Biomedical Technology (ICBBT), May 21-23, 2021*, Xi'an, China, in press.

[136] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[137] P. Svoboda and A. Di. Cara. Hairpin RNA: a secondary structure of primary importance. *Cellular and Molecular Life Sciences*, 63(7-8):901–908, April 2006.

[138] A.A. Szewczak and P.B. Moore. The Sarcin/Ricin loop, a modular RNA. *Journal of Molecular Biology*, 247(1):81–98, March 1995.

[139] Roopa Thapar, Andria P. Denmon, and Edward P. Nikonowicz. Recognition modes of RNA tetraloops and tetraloop-like motifs by RNA-binding proteins: Recognition modes of RNA tetraloops and tetraloop-like motifs. *Wiley Interdisciplinary Reviews: RNA*, 5(1):49–67, January 2014.

[140] Ignacio Tinoco and Carlos Bustamante. How RNA folds. *Journal of Molecular Biology*, 293(2):271–281, October 1999.

[141] Adrian Gabriel Torres, Eduard Batlle, and Lluis Ribas de Pouplana. Role of tRNA modifications in human diseases. *Trends in Molecular Medicine*, 20(6):306–314, June 2014.

[142] C. Tuerk, P. Gauss, C. Thermes, D. R. Groebe, M. Gayle, N. Guild, G. Stormo, Y. d'Aubenton Carafa, O. C. Uhlenbeck, and I. Tinoco. CUUCGG hairpins: extraordinarily stable RNA secondary structures associated with various biochemical processes. *Proceedings of the National Academy of Sciences*, 85(5):1364–1368, March 1988.

[143] Lee E. Vandivier, Stephen J. Anderson, Shawn W. Foley, and Brian D. Gregory. The conservation and function of RNA secondary structure in plants. *Annual Review of Plant Biology*, 67(1):463–488, April 2016.

[144] Vladimir Vapnik. *The nature of statistical learning theory*. Springer, 1995.

[145] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[146] Gabriele Varani and William H McClain. The G·U wobble base pair: A fundamental building block of RNA structure crucial to RNA function in diverse biological systems. *EMBO reports*, 1(1):18–23, July 2000.

[147] M. Vollmer, P. Sodmann, L. Caanitz, N. Nath, and L. Kaderali. Can supervised learning be used to classify cardiac rhythms? In *2017 Computing in Cardiology (CinC)*, volume 44, pages 1–4, 2017.

[148] E.Gerhart H Wagner and Klas Flärdh. Antisense RNAs everywhere? *Trends in Genetics*, 18(5):223–226, May 2002.

[149] Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, January 2009.

[150] Katherine Deigan Warner, Christine E. Hajdin, and Kevin M. Weeks. Principles for targeting RNA with drug-like small molecules. *Nature Reviews Drug Discovery*, 17(8):547–558, August 2018.

[151] Stefan Washietl, Ivo L Hofacker, Melanie Lukasser, Alexander Hüttenhofer, and Peter F Stadler. Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome. *Nature Biotechnology*, 23(11):1383–1390, November 2005.

[152] T.J. Wilson, M. Nahas, T. Ha, and D.M.J. Lilley. Folding and catalysis of the hairpin ribozyme. *Biochemical Society Transactions*, 33(3):461–465, June 2005.

[153] John C. Wootton and Scott Federhen. [33] Analysis of compositionally biased regions in sequence databases. In *Methods in Enzymology*, volume 266, pages 554–571. Elsevier, 1996.

[154] Cheng-Yan Wu, Qian-Zhong Li, and Zhen-Xing Feng. Non-coding RNA identification based on topology secondary structure and reading frame in organelle genome level. *Genomics*, 107(1):9–15, January 2016.

[155] Jacqueline R. Wyatt, Joseph D. Puglisi, and Ignacio Tinoco. RNA pseudoknots. *Journal of Molecular Biology*, 214(2):455–470, July 1990.

[156] Kaichao You, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. How Does Learning Rate Decay Help Modern Neural Networks? *arXiv:1908.01878 [cs, stat]*, September 2019. arXiv: 1908.01878.

[157] Malik Yousef, Waleed Khalifa, İlhan Erkin Acar, and Jens Allmer. MicroRNA categorization using sequence motifs and k-mers. *BMC Bioinformatics*, 18(1):170, December 2017.

[158] Mengge Zhang, Lianping Yang, Jie Ren, Nathan A. Ahlgren, Jed A. Fuhrman, and Fengzhu Sun. Prediction of virus-host infectious association by supervised learning methods. *BMC Bioinformatics*, 18(3):60, March 2017.

[159] Qin Zhao, Hung-Chung Huang, Uma Nagaswamy, Youlin Xia, Xiaolian Gao, and George E. Fox. UNAC tetraloops: to what extent do they mimic GNRA tetraloops? *Biopolymers*, 97(8):617–628, August 2012.