RNA Folding: Beyond the Thermodynamic Hypothesis

Max Ward (20748588)

*This report is submitted as partial fulfilment
of the requirements for the Honours Programme of the
School of Computer Science and Software Engineering,
The University of Western Australia,
2014*

Abstract

Algorithmic prediction of Ribonucleic Acid (RNA) secondary structure has been actively researched since the 1970s. RNA is a biologically active molecule with many poorly understood functions. For example, it is involved in regulation of DNA expression, has been implicated in developmental pathways, and acts as a catalyst for many biological processes. Quick and accurate prediction of RNA structure is therefore essential for further elucidation of its role. The Thermodynamic Hypothesis, which is the fundamental dogma underlying protein folding algorithms, has been applied successfully to the RNA folding problem. Unfortunately, this success is only partial; RNA folding algorithms are currently not able to reliably predict correct structures. These algorithms typically globally optimize some scoring function, usually thermodynamic stability. This is in keeping with the Thermodynamic Hypothesis. However, there is evidence that some RNAs fold into suboptimal states. One possible explanation is kinetic folding, which posits that structures form during transcription. In this investigation I hypothesize that local interactions are stronger than global interactions during RNA structure formation. To test this, a sliding window was used to generate locally optimal structures, then various algorithms were devised to merge these structures. An array of window sizes was tried, and the best were recorded. Given the right window size, the resulting predictions were more accurate than corresponding globally optimal structures, using the same model of RNA folding. I argue that this constitutes strong support for my hypothesis. In addition, a new algorithm called ‘*ab*-splat’, which was based on the computation of locally optimal windows, is introduced. This algorithm, while naive, had comparable prediction accuracy to the RNAfold algorithm, which is part of the ViennaRNA suite. Additionally, it runs an order of magnitude faster, and uses less memory.

Keywords: Ribonucleic acid, secondary structure, prediction
CR Categories: J.3 Biology and genetics

Acknowledgements

I would like to thank my mother, Virginia Ward, for providing half of my own DNA, and for the endless cups of tea. My thanks also go to Stephen Scourfield, for the similarly endless proof-reading he did of this dissertation. Amitava Datta also deserves praise for allowing me to pursue my own project, and for being a great sounding board, in addition to supervising my project. Finally, my thanks go to Eliot Courtney for helping with the time complexity analysis of *ab*-splat.

Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

The purpose of my investigation was to improve algorithmic Ribonucleic Acid (RNA) prediction. My report has been broken into five chapters. This first chapter comprises introductory information, an explanation of what Deoxyribonucleic Acid (DNA) and RNA are, the motivations for RNA structure prediction, and a definition of the problem domain. My hypotheses and the goals of my investigation are also stated in Chapter 1. The second chapter covers relevant literature regarding algorithms used for RNA structure prediction, and also emphasises their shortcomings. In Chapter 3, I present novel algorithms to test my hypotheses, and to predict RNA secondary structures. Following this, in Chapter 4, a detailed report of the testing procedure and recorded results is given. In Chapter 5, the final chapter, I discuss the implications of my findings, suggest future directions for more research, and state my conclusions.

1.1 DNA & RNA

DNA is the basic genetic building block upon which the classification of genetic material into genes and chromosomes is based. The role of DNA as the hereditary unit of genetics was determined in the 1940s [1]. Soon thereafter, Watson and Crick [44] published a highly acclaimed paper describing the fundamental chemical structure of DNA. In it, they outlined a double helix formation which has since become as iconic as it is canonical (see Figure 1.1). Each strand of the helix Watson and Crick discovered is essentially a chain of nucleotides. Each nucleotide is made of a sugar-phosphate backbone attached to a single base. The bases of corresponding strands form hydrogen bonds which hold the double helix together. The most astonishing and important of their findings was that these bases bond in a reciprocal fashion.

The reciprocal bonding relationships between bases is what allows replication to occur; a copy of DNA can be made by allowing the correct bases to bond to one of the strands making up its helix. This gives a model for inheritance

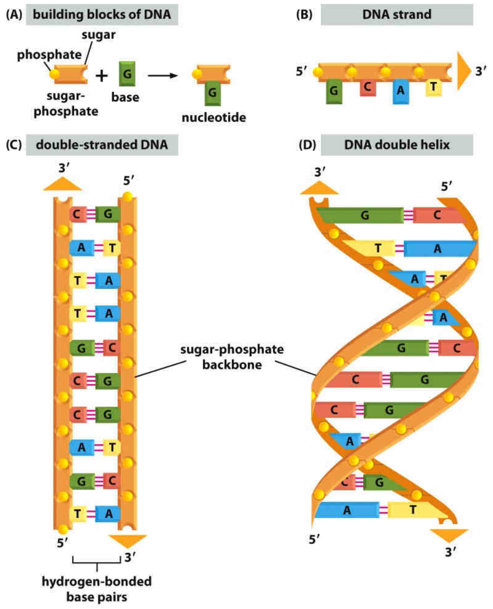

Figure 1.1: The structure and composition of DNA. Diagram taken from *Essential Cell Biology* [1].

and cellular replication. However, there remains the question of how DNA can actually code for protein. Proteins consist of amino acids bonded in a specific sequence [1]. The DNA must therefore code for amino acids. This code, which can be thought of as the 'digital' representation for the 'analogue' protein used by our cells, needs to be carried to ribosomes which translate it into protein [1]. This is a task carried out by RNAs.

As the name implies, RNA is chemically similar to DNA. However, unlike DNA, RNA is single stranded. This means that it is made of a single sequence of connected nucleotides. These nucleotides are like links in a chain. Each such nucleotide is attached to its direct neighbours in the sequence. Much like an actual chain, a complete RNA sequence is flexible, and can move about freely. While the nucleotides I have mentioned are chemically ingenious, we need not know the details. All that need be said here is that there are four types: Adenine (often abbreviated to A), Guanine (G), Cytosine (C), and Uracil (U). Adenine readily forms chemical bonds with uracil, and vice-versa. Similarly, guanine and cytosine have a propensity to bond. Additionally, guanine and uracil sometimes form weak bonds. In the parlance of RNA research, these types of bonds are called 'A-U', 'G-C', and 'G-U' respectively. RNA bonds to DNA and, in a sense, reads it. This results in the production of a copy of the DNA's genetic payload.

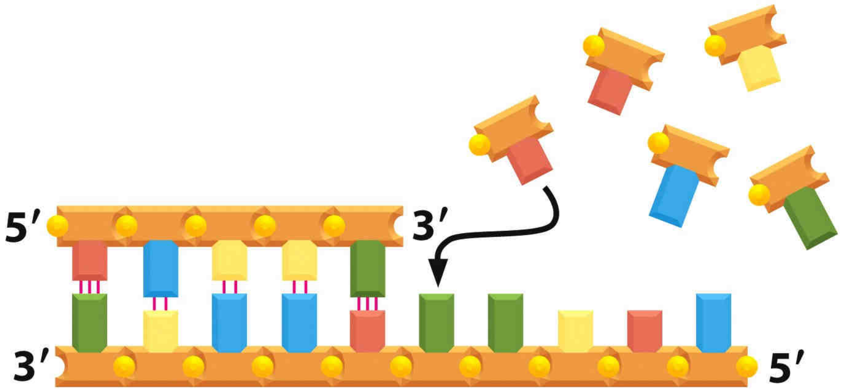

Figure 1.2: RNA transcription. Diagram taken from *Essential Cell Biology* [1].

This ‘downloaded’ information is then carried away to be translated into protein [1]. An example of this is depicted in Figure 1.2, in which we see a Messenger RNA molecule bonding to and thus making a copy of a section of DNA. As depicted in Figure 1.2, the 3’ end of a DNA or RNA molecule is the end onto which new nucleotides are added. The 5’ end is chemically stable, and nucleotides are not usually appended to it [1].

For many years the conventional wisdom was that DNA contained genes which coded for functional proteins used by the cell [1]. Though this is undoubtedly true, there was a problem: much of the human genome, and the genomes of other species, contains DNA which does not appear to code for anything [6]. Many theories have been put forward to explain this. It was argued that this ‘junk’ DNA is the perennial build-up of mutation, and that natural selection simply cannot act with strong enough selective force to cull this free-loading DNA [6]. Surprisingly, much of this non-coding DNA is transcribed into RNA, despite having no apparent function [21].

As it turns out, RNA is more than a simple messenger for encoded proteins. Recent research has found myriad important functions for RNA. For example, RNA can act as a catalyst for mRNA splicing and peptide bond formation, and can also alter the regulation of genes [46]. It seems that much of our genome contains templates for non-coding RNAs (ncRNAs). Amaral et al. [2] described our genome, and those of other eukaryotes, as being driven by a RNA machine. They noted that most of the eukaryote genome is transcribed into RNA, despite little of it coding for protein. These RNAs can interact with DNA, affecting gene expression. This allows DNA to regulate itself. For example, Makeyev and Maniati [24] reported that microRNAs affect the expression of genes by interfering with the translation of protein. They also argued that microRNAs, and other regulatory RNAs, explain the vast differences between organisms with similar genomes. To put this idea into perspective, we share roughly 90% of our genes with the domestic cat [31]. Mattick [25] has suggested that the process of development—from embryo to adult—is encoded in the interactions of such RNAs. Because of its inherently single stranded nature, RNA forms bonds with itself, folding into secondary and tertiary structures [9].

It is axiomatic that chemical structure is tantamount to biological function; RNA is no exception to this rule. For this reason, there has and continues to be an intense interest in predicting the secondary and tertiary structure of RNA molecules. This is in part because it will elucidate the underlying principles of RNA structure formation and function [9], but also because it will allow the detection and classification of unknown RNAs, enable prediction of novel RNA function, and assist the design of new RNA based drugs [8]. In fact, RNA is an extremely versatile molecule, and as such is attractive from both an engineering and computational point of view. Small combinatorial computational problems have been solved by representing the solution set using RNAs. Furthermore, a theory of computation has been put forward using self assembling RNA molecules [8]. As if to comment on the upheaval of a protein-centric view of biology in recent years, researchers have found that RNA is capable of supporting all the processes required for life without proteins [8]. The secondary structure of RNA is also highly conserved during evolution, indicating its importance [14].

Tertiary structure builds upon secondary structure, hence secondary and tertiary structures can be treated hierarchically [41]. As a result, it is possible to predict the secondary structure of a RNA without understanding the tertiary structure. This paper will focus on secondary structure prediction.

It holds to reason that an algorithm for RNA secondary structure prediction can never be realised if we do not understand how these structures form, or their general morphology. For this reason it is important to understand how true RNA secondary structures can be determined, and the limitations of these techniques. DNA and RNA molecules can be analysed using X-ray crystallographic methods. These types of approaches work because the wavelengths of some X-rays are the same as the dimensions of DNA and RNA inter-atomic bonds. The diffraction of X-ray light by these molecules can thus be observed, and their structures can subsequently be inferred by analysis of the resulting data. Nuclear Magnetic Resonance (NMR) is another technique which can be applied to the analysis of DNA or RNA. It relies on the spin of atoms when in a magnetic field. These spin signals can be used to determine the atomic composition and topology of a

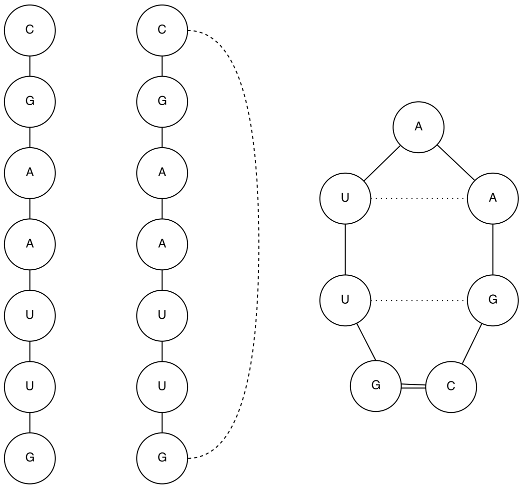

Figure 1.3: A simple example of how a RNA sequence might fold. Dotted lines represent potential bonds between nucleotides, double lines represent actual bonds. The progression of folding follows from left to right.

molecule. This has the advantage of not requiring the molecule under observation to be crystallized before analysis. Arguably NMR gives a better in vivo view of RNAs and DNAs, which are fundamentally flexible structures. NMR also has some disadvantages; for instance, it is less accurate than X-ray crystallography, and cannot be used on extremely large molecules. The reason these techniques cannot be used for all RNA structural assays is that they are extremely expensive and time consuming. For more information refer to *Principles of Nucleic Acid Structure* by Stephen Neidle [28].

RNA secondary structures are a lot like paper aeroplanes. When constructing a paper aeroplane, one starts with a blank sheet of paper, and folds it repeatedly upon itself. This is like what happens when RNA molecules form bonds. Say a G and C nucleotide began forming a chemical bond. This would cause the entire nucleotide chain to fold on itself as the G and C nucleotides come into close proximity. After the bond has formed, we are left with a loop containing the nucleotides that were between the G and the C in our theoretical RNA (see Figure 1.3). The folding process may then repeat itself for nucleotides that are contained inside this loop region. Hence we say that the bonds of RNA molecules are nested. Folds form within folds; the structure of RNA is recursive. To clarify these concepts, let us recapitulate the paper aeroplane analogy. Paper aeroplanes are made from finite, two dimensional rectangles that fold repeatedly in three dimensions. In contrast, RNA molecules are finite, one dimensional line segments that fold repeatedly in two dimensions. More detail can be found in "How RNA Fold" which was published in the *Journal of Molecular Biology* [41].

RNA secondary structure prediction techniques can be broadly broken into two categories: those that use auxiliary information to assist in prediction, and those that predict structure ex nihilo—that is, with nothing but the 'proband' sequence we require a structure for. The former approach typically entails consensus matching between some sequences for which a user already knows the secondary structures, and a sequence for which the structure is unknown [14]. In this paper I investigate the latter approach. This class of prediction algorithm has been actively researched since the 1970s. Despite this, current techniques are often unreliable, and are unable to accurately predict the structures of some RNAs. Unfortunately these are the most generally useful algorithms as they can predict RNA structure given only the RNA sequence itself. This may be because they attempt to globally optimize some scoring function; typically the scoring system reflects thermodynamic stability. The core hypothesis of this investigation is that local interactions can be stronger than global interactions during RNA structure formation. Additionally, modelling such interactions may improve prediction accuracy. If these hypotheses are supported, it follows that improvements in algorithmic prediction of RNA secondary structure can be achieved.

CHAPTER 2

Relevant Literature

2.1 Dynamic Programming Techniques

2.1.1 The Nussinov Algorithm

The first such algorithms were based on relatively naive brute force. All possible secondary structures are enumerated, and the one with the most bonds is selected as the solution [30]. While being simplistic these first approaches introduce an important assumption: RNA molecules will form energetically stable secondary structures. Maximising bonds is a crude but nonetheless accurate measure of energetic stability, as every bond increases the stability of a structure [30]. In the late 1970s, when the first large RNA molecules were being successfully sequenced, Nussinov et al. [30] introduced an algorithm based on loop matching for bonding pairs. Their algorithm finds a single structure having maximal bonds, with the restriction that all bonding pairs must be entirely nested. It does this in $O(N^3)$ time and using $O(N^2)$ space by utilizing dynamic programming techniques. Thence, Nussinov and Jacobson [29] introduced a refined version of the same algorithm, and began testing it against experimentally verified RNA secondary structures. They had mixed success; transfer RNAs (tRNAs) proved conspicuous in their difference from predicted structures.

Because of its dynamic programming nature, this algorithm performs recursive decompositions of the RNA, building larger structures out of repeated substructures. A natural representation of this is depicted in Figure 2.1. Part A of Figure 2.1 shows bonds as arcs across a circular graph. In it, we see the nested nature of the structures being explored by the Nussinov algorithm. Part B shows how these structures translate to actual RNAs in vivo. It also introduces the standard decompositions of secondary structures, namely the hairpin loop, the interior loop, and the branch junction or multiloop. Unlabelled in the diagram are stems; these are stacked base pairings, for example $B1$, $Bn-1$ and $B2$, $Bn-2$.

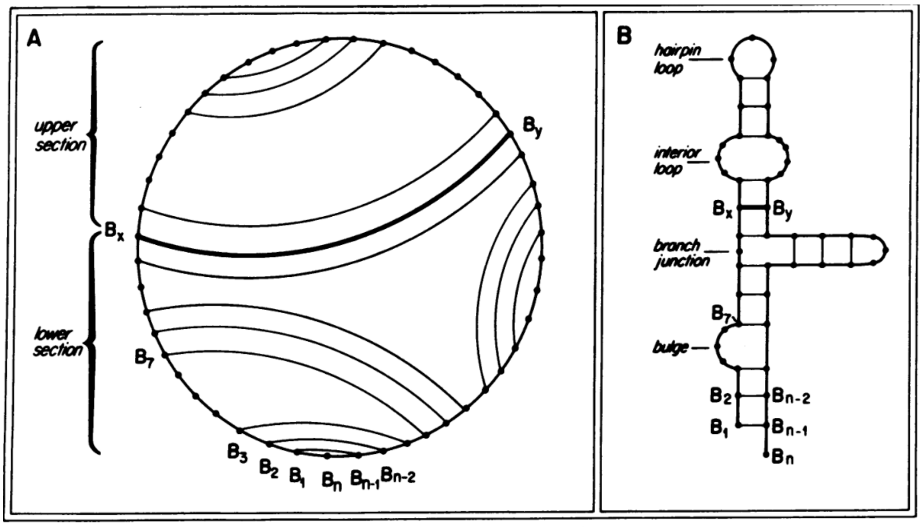

Figure 2.1: RNA secondary structure as described in the Nussinov algorithm. Taken from the original publication [29].

$$\begin{aligned} M(i,j) &= \max\{A, B, C, D\} \\ A &= M(i, j-1) \\ B &= M(i+1, j) \\ C &= M(i+1, j-1) + W(i,j) \\ D &= \max\{M(i,k) + M(k+1,j)\}\ \mathit{when}\, i < k < j \end{aligned} \tag{2.1}$$

The recurrence relation for the Nussinov algorithm is defined in Equation 2.1. In it, the bases of an RNA are referred to by the indexes i and j. This is the typical notation for RNA folding algorithms. If one imagines the RNA being laid out lengthwise from the 5' end to the 3' end, index `0` refers to the left most base, its right hand neighbour is index `1`, and so on. The first two cases (A and B) find the score associated with not allowing the bases corresponding to indexes i and j to bond. Case C conversely determines the score given that i and j are bonded. The final case D computes the score associated with a bifurcation. A bifurcation is the decomposition of RNA into two separate structures. This recurrence relation implies a $O(N^3)$ worst case time complexity and a $O(N^2)$ space complexity, as a $O(N^2)$ state space (all combinations of i and j) is explored with a linear time recurrence relation. In the original algorithm the minimum size of hairpin loops was limited, as real RNAs typically do not have hairpin loops fewer than three bases in size. The recurrence relation presented here has also been modified for the sake of clarity (cases A and B can be merged into case D), but the logic of the algorithm is equivalent.

This algorithm can also be extended to accommodate a more advanced energy model. Instead of weighting each bond equally, bonds can be weighted according to the proportion they are expected to contribute to the molecule's stability [29]. When considering the value of a bond, it might be given greater weight if it adds to the formation of a stem (a stabilizing structure), or given lower weight if it forms an internal loop or bulge, as these generally destabilize RNA molecules [29]. Unfortunately it is hard to find good values for such weights, and determining which substructure a bond contributes to requires backtracking. Additionally it should be noted that the Nussinov algorithm is old technology, and is no longer used for the prediction of RNA secondary structures. I have presented it in detail because it forms the basis for the Zuker algorithm, which will now be fully discussed.

2.1.2 The Zuker Algorithm

Soon after the work of Nussinov and Jacobson, Zuker and Stiegler [48] described an altered version of the same algorithm which, instead of maximising base pair weights, minimizes the free energy of secondary structures. This is done by introducing a number of thermodynamic rules for canonical substructures like hairpin loops, internal bulges, multiloops, unbonded base pairs, and stacked base pairs. The algorithm is similar to the Nussinov algorithm, but requires another mutually recursive dynamic programming recurrence to inject a relatively comprehensive thermodynamic scoring system. The original thermodynamic scoring scheme is borrowed from the work of Studnicka et al. [39] who presented a complex but theoretically similar algorithm, albeit with much worse asymptotic and implementation complexities.

Before describing it in detail I shall first introduce some useful terminology, which should clarify aspects of Zuker and Stieglers algorithm. The bases of a RNA molecule can be thought of as vertices in a planar graph. Edges between vertices may then be represented as chords on a semicircular diagram (see Figures 2.1 and 2.2). These chords are not allowed to touch. A chord is admissible if it represents a chemically valid bond, and an admissible structure is a structure whose graph contains only admissible bonds. Thus, one can define a face of such a graph as any planar region bounded on all sides. The folding algorithm of Zuker and Stiegler considers such faces as the basic contributing factor to a molecule's

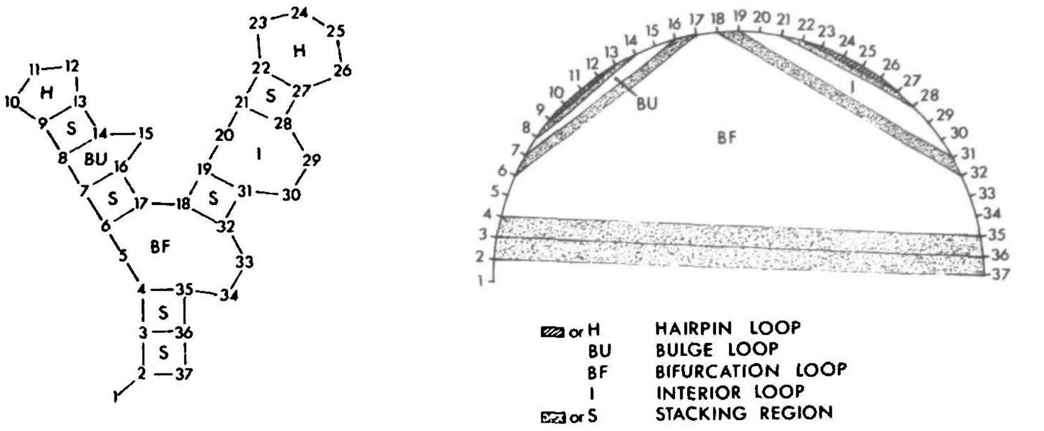

Figure 2.2: Diagram of faces used in the Zuker algorithm. Taken from original publication [48].

stability, unlike the algorithm of Nussinov and Jacobson which considers only individual bonds.

Let $E(F)$ represent the energy of a face F; impossible structures are given an energy value of infinity, for example, hairpin loops smaller than three bases. In addition let $V(i,j)$ be defined as the minimum free energy given that bases i and j are bonded, and let $W(i,j)$ represent the minimum free energy of all structures contained within bases i and j inclusive. Note that for $W(i,j)$ there need not be a bond between bases i and j. Also, if i and j cannot bond then $V(i,j)=\infty$. Finally note that $FH(i,j)$ represents a hairpin loop structure from i to j, and that $FL(i,j,i',j')$ is defined as the face bounded by the bonds i,j and i',j'. Examples of these decompositions are shown diagrammatically in the right half of Figure 2.2. The labelled regions show faces in a semicircular graph representing a strand of RNA. In the accompanying left half of the figure, the same structure is shown as it would appear in a real RNA.

$$\begin{aligned} V(i,j) &= \min\{E1, E2, E3\} \\ E1 &= E(FH(i,j)) \\ E2 &= \min\{E(FL(i,j,i',j')) + V(i',j')\}\ \mathit{where}\, i < i' < j' < j \\ E3 &= \min\{W(i+1,i') + W(i'+1,j-1)\}\ \mathit{where}\, i+1 < i' < j-2 \end{aligned} \tag{2.2}$$

As shown by the definition provided in Equation 2.2, $V(i,j)$ is computed by minimizing three cases. The first case considers the bond between i and j closing off a hairpin loop (H in Figure 2.2). The second accounts for situations in which the bond from i to j results in a bulge (BU in Figure 2.2), internal loop (I in Figure 2.2), or the continuation of a stacking region with the interior bond i', j' (S in Figure 2.2). The third and final case considers possible bifurcations contained within the bond between i and j (BF in Figure 2.2).

$$\begin{aligned} W(i,j) &= \min\{W(i+1,j), W(i,j-1), V(i,j), E4\} \\ E4 &= \min\{W(i,i') + W(i'+1,j)\}\ \mathit{where}\, i < i' < j-1 \end{aligned} \tag{2.3}$$

Equation 2.3 is the recurrence for $W(i,j)$ as described by Zuker and Stiegler. Again there are three cases. The first two cases, $W(i+1,j)$ and $W(i,j-1)$, should be thought of together, and account for possibilities having no bond between i and j. This is similar to cases A and B from the Nussinov algorithm (Equation 2.1). The third case considers taking the bond from i to j. The fourth and final case allows for bifurcations in which two bonding regions split the structure into two sections. The final minimum free energy of the best structure is defined by $W(0, n-1)$, where n is the length of the RNA molecule. It should be noted that the free energy for small structures (fewer than 6 nucleotides in length) can easily be precomputed, and forms the base case of the given recurrence relations. Because of its efficiency ($O(N^3)$ time and $O(N^2)$ space), robustness, and extensibility, this method is, even today, still the most popular available. The most widely used packages for RNA secondary structure prediction all contain implementations of the Zuker algorithm. Some notable examples are RNAfold from the ViennaRNA package [22], Zuker's own implementation Mfold [47], and the recently updated RNAstructure [7]. The Zuker algorithm suffers a major shortcoming, however. Because all bonding regions are assumed to be nested, it cannot handle the case of pseudoknots.

2.2 Pseudoknots

Pseudoknots are structures in which a bonding pair may have its first base inside another bonding pair, and the other base outside that bonding pair. In short, it is not properly nested. These structures are not common, but have been experimentally verified in numerous RNAs [40]. Additionally, pseudoknots also appear to perform useful biological functions. For example, pseudoknots have been shown to allow frame shifting during translation of proteins [27]. In layman's terms, pseudoknots can change the way RNA is read when being translated into protein. Frame shifting is used extensively by viruses, particularly HIV [27]. Unfortunately, the problem of finding optimal structures with pseudoknots has been shown to be NP-Complete [23]. In my investigation I did not consider the prediction of pseudoknotted RNAs as they are uncommon and difficult to model. However, it is important to understand how such structures could be integrated into the algorithms presented in this paper. As such, I have included a succinct overview of pseudoknot prediction techniques.

Despite the problem being NP-Complete, in 1999 Rivas and Eddy [35] introduced an ingenious dynamic programming algorithm based on a new thermodynamic model encompassing pseudoknots. Their algorithm can predict a large (but incomplete) set of pseudoknot classes using $O(N^6)$ time and $O(N^4)$ memory. They generalised the Zuker method by using a gap matrix to represent multiple regions being considered for bonding, rather than the single continuous region used in the Zuker method. Because of its extreme space and time requirements this algorithm is used only sparingly in practice; other thermodynamic based methods for pseudoknot prediction have been formulated using similar principles. Deogun et al. [11] described an algorithm which can handle a restricted class of pseudoknots (only non-recursive pseudoknots) in $O(N^4)$ time and using $O(N^3)$ space. Shortly after this, Reeder and Giegerich [33] presented an algorithm that can predict only simple recursive pseudoknots that meet their 'canonization' criteria, and which requires $O(N^4)$ time and $O(N^2)$ space. While seemingly restrictive, this does, in fact, predict a large array of pseudoknots accurately.

In recent years different approaches have been explored. In 2010 Sperschneider and Datta introduced DotKnot [37], which improves upon previous algorithms by using probability dot-plot guided heuristics, and an updated energy model, to predict pseudoknots. DotKnot is able to predict pseudoknots more accurately than existing methods with more frugal space and time requirements. This technique was later refined so that it could predict H-type pseudoknots and intramolecular kissing hairpins [38].

2.3 Accuracy

It is important to test and compare the accuracy of various prediction methods. As such, well established techniques have been developed over the history of RNA structure prediction. Usually accuracy is determined by comparing predicted structures to known structures. True Positives (TP) is defined as the number of base pairs which appear in both the predicted structure and the actual structure. False Positives (FP) is the number of predicted base pairs not in the true structure [22]. Similarly, False Negatives (FN) is defined as the number of base pairings in the reference structure but not present in the predicted structure [22]. Sensitivity, also called the True Positive Rate (TPR), can be defined using the previously introduced values. I have given a mathematical definition of TPR in Equation 2.4.

$$\frac{TP}{TP + FN} \tag{2.4}$$

Precision, sometimes known as Positive Predictive Value (PPV), can also be calculated using these values (see Equation 2.5).

$$\frac{TP}{TP + FP} \tag{2.5}$$

RNAfold [22] is one of the leading RNA folding algorithms, and is made available as part of the ViennaRNA package [22]. At its heart, it is an implementation of the original dynamic programming algorithm first discovered by Zuker, albeit with a more refined energy model. It is an extremely efficient implementation of this algorithm, and is also one of the most accurate in terms of sensitivity and precision as compared to other implementations of the same algorithm [22]. When Reeder and Giegerich [33] first described their algorithm (implemented in the pknotsRG package) for pseudoknot prediction they compared it to RNAfold, and the algorithm of Rivas and Eddy [35] (implemented in the same package and hereafter referred to as pknotsRE). Their algorithm generally had higher sensitivity than both other methods, but it is worth noting that pknotsRE was comparable, despite being based on an outdated energy model. This is possibly explained by the fact that it is a more general, and thus more powerful, algorithm. RNAfold lagged behind pknotsRE and pknotsRG in sensitivity, but executed orders of magnitude faster. Indeed, it has been shown to have state of the art accuracy for RNAs containing no pseudoknots while also exhibiting unrivalled computation speed [22].

2.4 Local Structure Prediction

DNA sequences, unlike typical RNA sequences, are very large indeed, usually hundreds of megabytes of data. DNA sequences often contain subsequences that code for RNAs. Functionally important RNAs typically have a recognizable

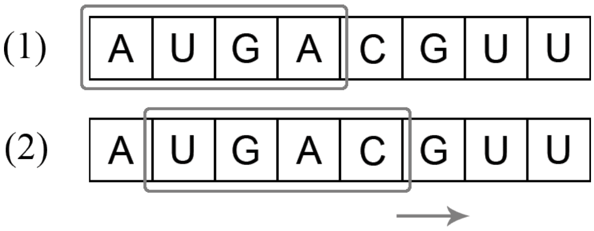

Figure 2.3: Depiction of how sliding windows can explore a RNA sequence.

secondary structure. When searching a large genome for functional RNAs, one can use a sliding window of fixed size to find locally optimal structures (see Figure 2.3). This can be done by running a typical cubic time implementation of the Zuker algorithm at every window location. Let L be defined as the chosen window size, and N represent the length of the genome. This leads to a total complexity of $O(NL^3)$. While not prohibitive, this becomes intractable for many genomes, which are typically extremely large—millions or billions of bases.

In 2004, Hofacker, Priwitzer, and Stadler [15] provided an excellent insight, and lowered this bound to $O(NL^2)$. It is thus possible to scan large genomes for interesting RNA secondary structure motifs. This is achieved by modifying the Zuker algorithm. Specifically, the dynamic programming table from the previous step is used to quickly fill the table for the next window in quadratic time; because consecutive windows overlap, preceding information can be meaningfully used in each forward computational step. As a result it requires only a single table of size $O(L^2)$, and as such its memory complexity is only $O(N+L^2)$. Later, in 2009, Horesh et al. [16] managed to lower the expected time bound to $O(NL)$ under the assumption that one is folding RNAs that are typical of naturally occurring sequences. This average case time complexity has been experimentally verified, as their algorithm is shown to outperform that of Hofacker, Priwitzer, and Stadler.

Clearly, good algorithms are available for the folding of consecutive RNA windows. For even modest sized RNAs, such algorithms are orders of magnitude faster than holistic secondary structure prediction algorithms. However, this comes with the major caveat of not actually predicting a complete secondary structure, but only a set of locally optimal structures.

2.5 Context Free Grammars

RNA sequences and their secondary structures can be represented as Context Free Grammars (CFGs). Various production rules output different internal structures (such as hairpin loops, or internal bulges) as symbols, with terminals being bases A, U, G, and C. This is a fundamentally different approach to those discussed previously. However, CFGs can, in fact, use the same thermodynamic energy model. Stochastic Context Free Grammars (SCFGs) can be used to encode the plausibility of production rules, and thus find the most plausible parse tree using a thermodynamic model [36]. In addition, these kinds of algorithms can be trained to incorporate statistical information such as phylogenetic similarity, or machine learned parameters [36]. These kinds of methods have difficulties with pseudoknots, as non-nested structures are not compatible with CFGs. A notable workaround was applied by Kato, Seki, and Kasami [17], who used multiple context free grammars to model pseudoknots. However, their approach increases time and space requirements prodigiously.

The greatest strength of CFG based approaches is that they can diverge from the use of free energy minimisation entirely. This is advantageous as using a physics based model, such as free energy minimization, requires a large volume of experimentally verified parameters. For this reason, many parameters are often not included in such models because they cannot be quantified empirically. The energy value of multi-branch loops, for example, is not known, and is usually guessed in modern RNA prediction algorithms. Likewise, the inter-structural interactions of hairpin loops, bulges, multi-branch loops, and internal loops have not been quantified experimentally and is thus not used as a free energy parameter. CONTRAfold [13] was one of the first SCFG based algorithms to achieve comparable performance to Zuker-like free energy minimization methods. It does away with the notion of free energy minimization, and instead uses a set of trained parameters based on conditional log-linear models. CONTRAfold achieved an average prediction sensitivity higher than RNAfold [22], and also higher than that of Mfold [47].

2.6 Soft Computing

The use of soft computing techniques has also yielded some success in the prediction of RNA secondary structure. Koessler et al. [18] modelled RNA structures as a tree of internal substructures, then used artificial neural networks to recognize which of these trees appeared most RNA like. These trees are generated by constructing basic secondary structures and combinatorially merging them to form many trees, each of which is represented as a vector of simpler trees. This vector is used as the input to the neural network.

This kind of combinatorial blending of RNA stems is a technique shared by genetic algorithms. Indeed, this was precisely the starting point of Van Batenburg, Gultyaev, and Pleij [43], who used a simple genetic algorithm to predict secondary structure. Their algorithm starts by computing an array of all possible stems; each genome is represented as a binary string where `1` indicates a stem is in the candidate structure, and `0` indicates that it is not. Their genetic algorithm proceeds by seeding the genomes with random bits, then in a series of generation steps performs typical binary mutation, crossover, and breeding, conserving and selectively breeding the fittest solutions. Fitness is defined in their algorithm as the summed length (number of bonds) of all stems. In an improved version, summed stacking free energy reduction is used instead.

Unfortunately they discovered a problem with this approach: the population contained a relatively large portion of zero fitness individuals. This is because many combinations of stems are incompatible with each other, yielding impossible structures. Instead of giving these structures zero fitness, they altered their algorithm slightly to disallow crossover for stems that create an invalid structure. In addition to this, they also explored an important advantage of genetic algorithms for RNA secondary structure prediction: that of kinetic folding. Kinetic folding is the hypothesis that some RNAs, particularly large ones, have a rugged energy landscape, and because of the incremental process of transcription and folding become stuck in suboptimal areas during folding [42, 43]. The algorithm of Van Batenburg, Gultyaev, and Pleij simulated this process by limiting the size of stems that could contribute to a genome, and increasing this size over time until the length of the RNA was reached. This single modification to their algorithm yielded the greatest improvement in predictive power. It should also be noted that it can predict pseudoknots, as the algorithm does not force stems to be nested. Despite this, their approach was still less accurate than the dynamic programming approaches they compared it to. This is likely because their energy model was puerile in comparison, rather than because the algorithm is flawed.

Indeed, Wiese, Deschenes, and Hendriks [45] introduced an improved genetic algorithm based on the same principles as that of Van Batenburg, Gultyaev, and Pleij, but instead using an advanced energy model for fitness. They demonstrated that it outperforms the popular dynamic programming algorithm Mfold [47], which uses a similarly complex model. Wiese, Deschenes, and Hendriks noted that as the length of RNA molecules increases, the correlation between lower free energy and accuracy decreases. They concluded that this is due to the incompleteness of the current thermodynamic model of RNA folding. Unlike the algorithm of Van Batenburg, Gultyaev, and Pleij, their algorithm is oblivious to kinetic folding, and does not attempt to simulate it.

2.7 Kinetic Folding

Kinetic folding has been observed in vivo, and evidence has existed since the early 80s. Kramer and Mills [19] reported seeing preliminary secondary structures form, break apart, and reform into other structures during transcription. This means that as RNA is synthesized its structure is already undergoing dynamic formation with every additional nucleotide. Over a decade later, Morgan and Higgs [26] examined many RNA molecules and found that typical structures have suboptimal free energy. They postulated that this is due to kinetic folding, and that errors in the free energy parameters of the current model cannot fully explain the discrepancy. Recently, Proctor and Meyer [32] improved secondary structure prediction accuracy by simulating kinetic folding. In their method, the effects of kinetic folding are captured using a scaling function in which closer nucleotides are weighted higher than distant nucleotides. Proctor and Meyer [32] reported that this improves prediction accuracy; particularly for RNA molecules comprising greater than 1000 nucleotides.

2.8 State of the Art: Global Optimization

Most state of the art algorithms fold RNA in a way that globally maximises score according to some model. The Zuker algorithm, for example, finds the global minimum free energy configuration. SCFG based algorithms maximise the probability of a parse tree. This bias is largely due to the 'Thermodynamic Hypothesis'. Anfinsen [4] presented this hypothesis as the underlying principle for the formation of biologically active proteins. He held that proteins fold into a minimum Gibbs free energy conformation in their typical biological environment. (Environment being defined as the molecules' physiological state: pH, temperature, and ion concentration.) Furthermore, through natural selection, molecules that are most likely to fold into the correct shape have evolved. Therefore we should be able to determine the structure of biologically active molecules (particularly proteins in Anfinsen's original thesis) by finding a minimal free energy conformation, given the building blocks of the molecule. This insight has been invaluable for folding proteins and RNAs in silico. Despite this, it has recently become clear that methods for the prediction of RNA secondary structures have hit an upper limit in accuracy.

In her discussion of modern RNA prediction, Rivas [34] unifies pseudoknot-free RNA folding algorithms. Her core observation is that all such prediction algorithms contain the same four key components: an architecture, or the production rules of a grammar; a scoring scheme, or how scores are assigned to these production rules; and the parametrization of the scoring scheme, or the specific values assigned to it. These features are referred to by Rivas, and by me in the following discussion, as the 'model'. The fourth and final feature is the folding algorithm used to find the best structure given the model. Here Rivas notes that the two dominant folding algorithms are interchangeable. The Cocke-Younger-Kasami (CYK) algorithm, used to parse SCFGs, and free energy minimizing algorithms based on the work of Zuker, are isomorphic for the purpose of parsing RNA grammars. Rivas additionally notes that all scoring schemes and parametrizations appear to hit an accuracy upper limit, and that complex, machine learned models are only slightly more accurate than thermodynamic models. In fact, relatively basic grammars with hundreds of parameters seem to perform almost equivalently to those with tens of thousands. While Rivas managed to unify many aspects of RNA prediction she did not recognise that all such algorithms are based on those same assumptions underpinning the thermodynamic hypothesis. They seek to globally maximise a scoring function for the final RNA secondary structure.

I propose that, in RNA molecules, local interactions are stronger than global interactions. As a result, RNA molecules will misfold into a global structure made up of locally optimal structures. In other words, short range interactions will predominate long range interactions. This is my core hypothesis. If this assumption is correct, it follows that there exists a set of 'windows' that, when folded using any reasonable model, will be more accurate than the corresponding global optimum using the same model. There is already some evidence for this hypothesis. Dawson et al. [10] used variable Kuhn lengths to accurately predict RNAs containing less than 100 nucleotides. Kuhn length is the size of a segment in a polymer chain. It is a simplifying assumption that allows one to treat the entire chain as a sequence of Kuhn segments. In addition, Dawson et al. showed that the energy landscape of these RNAs, when the correct Kuhn length was applied, was funnel shaped. Without such simplifying assumptions the energy landscape of RNAs is notoriously rugged, with in vivo secondary structures often becoming 'trapped' in suboptimal states; this is most apparent in large RNAs [12].

If my hypothesis is supported, I aim to leverage locally optimal windows to improve the accuracy and speed of RNA secondary structure prediction.

CHAPTER 3

Locally Optimal Algorithms

This chapter contains descriptions of the novel algorithmic approaches applied in my investigation. First I clarify what local optimization means, and define some useful terms. Specifically, I shall explain how locally optimal structures are computed, and the terminology I use to describe components of this process. Thence I describe some algorithms which can be used to merge locally optimal structures into a global structure. Building upon these algorithms, I introduce *ab*-splat, a novel algorithm for the prediction of RNA secondary structures.

3.1 RNA Intervals

Running an algorithm which generates consecutive windows of size L over a RNA primary sequence of size n produces $n - L + 1$ windows of size L. Each of these windows contains the optimal secondary structure for that subsequence of RNA. This structure is computed with complete disregard for any interactions outside the bases encompassed by the window. It is therefore locally optimal, but not globally optimal. The secondary structure may contain no bonds at all, or it might contain an elaborate structure. Such a structure might have disjoint components. An example of this is shown in Figure 3.1, in which there are two distinct substructures which are part of no larger substructure. A substructure is disjoint if and only if no bond encompasses it and another substructure.

I extracted all disjoint substructures from the structures computed by the sliding window algorithm. I called these ‘RNA intervals’. This was done so that structures from different windows could be effectively blended. Every RNA interval was assigned a score. This score was defined as the amount of free energy reduction which that RNA interval contributes. Figure 3.1 contains a diagrammatic representation of both the sliding window, and the RNA intervals it comprises. In the diagram, only a single window is shown, however the same logic holds for all previous and subsequent windows. RNA structure is represented using dot-bracket notation. In this style, every matching pair of parentheses

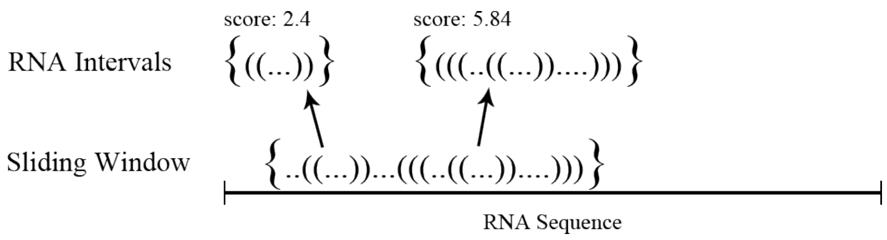

Figure 3.1: The relationship between a sliding window and its RNA intervals. RNA secondary structures are represented using dot-bracket notation.

represents bonded base pairs. Conversely, a full stop represents an unbonded base. In the figure, curly braces indicate the ends of windows and RNA intervals. The computed structure for the window contains two disjoint substructures, these are split into two RNA intervals. These are disjoint because they are contained within no other bond. The RNA interval merging algorithms, which I shall now discuss, worked on these disjoint substructures, rather than on the entire structures produced by the sliding window.

3.2 Merging RNA Intervals

Given a RNA sequence, one can generate a set of RNA intervals by running any sliding window algorithm, and storing the RNA intervals generated for each window location. Many of these intervals will overlap, or contain fragments of one-another. To construct a plausible and complete secondary structure for the original RNA, a subset of the RNA intervals generated this way must be selected. I devised several methods for doing this. The goal of such an algorithm is to produce a valid secondary structure out of a set of RNA intervals which is as accurate (compared to the actual secondary structure) as possible.

Some definitions are provided here to elucidate concepts discussed in the following section. Let W be the set of all RNA intervals to choose from, and let the set S represent the set of selected intervals. Also, an interval is compatible with another if they do not overlap and neither interval contains the other (see Figure 3.2). Taking only mutually compatible intervals will result in a valid RNA structure. The way in which these intervals are selected determines the accuracy of the resulting structure. I refer to this class of algorithms as 'RNA interval merging algorithms'. If any RNA interval merging algorithm could find RNA secondary structures which are consistently more accurate than those generated by conventional prediction algorithms, my hypothesis would be supported. This

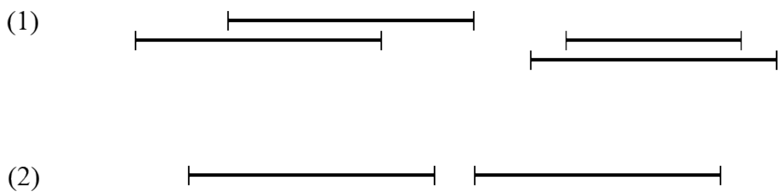

Figure 3.2: Case 1 shows RNA intervals that are not compatible. Case 2 shows an example of compatible intervals.

is because such an algorithm builds final structures out of a collection of locally optimal substructures. In other words, only local interactions are used to predict the resulting structure.

Top-Down Selection

In this method, the set W is sorted by RNA interval size (number of bases) in descending order. The algorithm then examines each element of W in order. If an element of W is compatible with all elements of S, it is added to S.

Algorithm 1 Top-Down Selection

```
Let $W$ be the set of RNA intervals
Let $S = \{\}$
Sort($W$)
for all $W$ as $e$ do
    if Compatible($e$, $S$) then
        $S = S \cup e$
    end if
end for
return $S$
```

The algorithm (see Algorithm 1) examines every element of W exactly once, and for each element checks for compatibility with S. The naive way to check for compatibility is to examine every element in S. This would give the algorithm a worst case time complexity of $O(n^2)$, where n is the number of intervals, as all the intervals in W may be mutually compatible. This is a result of the algorithm first checking S of size 0, then 1, up to $n-1$, totalling $(n-1)((n-1)+1)/2$ steps. Additionally it requires $O(n)$ space to store S.

It is possible to lower the time complexity to $O(n \log n)$ by storing the elements of S in an augmented interval tree. Querying this tree finds any elements of S

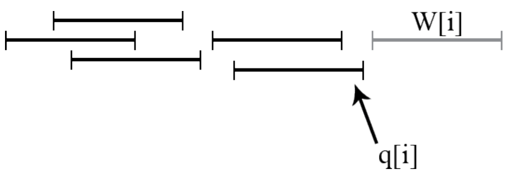

Figure 3.3: The interval stored in $q[i]$ for the RNA interval $W[i]$.

that intersect the interval in question. This query takes $O(\log n)$ time and is executed n times—once for every element of W. This is optimal, since the `Sort` operation requires $O(n \log n)$ time. Hence, the final worst case time complexity is $O(n \log n)$ and the final worst case space complexity is $O(n)$.

Bottom-Up Selection

This algorithm is identical to Top-Down Selection, except that W is sorted by interval size in ascending order.

Score Selection

This algorithm is identical to Top-Down Selection, except that W is sorted by score in descending order.

Weighted Activity Selection

The Weighted Activity Selection problem is a generalization of the activity selection problem. Given a set of intervals, each of which is assigned a weight, the problem is to find a subset such that none of these intervals touch one-another, and the sum of their weights is maximised. This maps naturally to the problem domain of this investigation. I used this algorithm to find the subset of compatible windows which have the minimum sum free energy, or put another way, maximum score.

The initial sort (see line 1 in Algorithm 2) requires $O(n \log n)$ time. In the following loop (line 4) the algorithm finds the right most element in W such that it is compatible with element i (see Figure 3.3). This can be done using a binary search. Hence the total worst case run time of the loop is also $O(n \log n)$. The final dynamic programming array fill (line 9) requires only $O(n)$ time. As a result the final time complexity of this algorithm is $O(n \log n)$. Additionally, $O(n)$ space is required for the arrays q and dp.

Algorithm 2 Weighted Activity Selection

```
Sort $W$ by right end points
Let $n = |W|$
Let $q$ = array[1..$n$]
for $i = 1 \to n$ do
    $q[i]$ = highest index $< i$ which is compatible with $i$
end for
Let $dp$ = array[0..$n$]
$dp[0] = 0$
for $i = 1 \to n$ do
    $dp[i] = \max(weight[i] + dp[q[i]],\ dp[i-1])$
end for
return $dp[n]$
```

3.2.1 Closing Remarks

The RNA interval merging algorithms discussed all have the same $O(n \log n)$ worst case time complexity, and the same auxiliary space complexity of $O(n)$. None of these algorithms have large constant factors. Therefore, my final choice of algorithm was based solely on the accuracy of computed secondary structures. Additionally, I wish to make clear that these algorithms are not novel in isolation. It is their application to RNA folding that I claim is novel. The Weighted Activity Selection algorithm in particular was not created by me; it is a canonical dynamic programming algorithm. However, I could find no credited inventor for it.

3.3 Prediction Using Windows

RNA interval merging algorithms are not able to make good predictions in isolation. Hence, an approach was devised to predict RNA secondary structures using them. In this approach I computed sliding windows for a set of sizes—a sample like this is often called a 'splat'. Weighted Activity Selection was then done on the resulting set of computed RNA intervals. This was because it appeared to be the best RNA interval merging algorithm, see Section 5.1 for details. I explored several methods for finding good splats. I present here the most successful technique, which I call '*ab*-splat' prediction. Initially the algorithm starts at a small window size, and exponentially increases the scope of the window until a threshold is exceeded. This is explained using pseudocode in Algorithm 3.

The `Threshold` function was defined as $\sqrt{\texttt{RNA Length}} \times 9.5$. This formula

Algorithm 3 *ab*-splat

```
Let $RNA$ be the primary RNA sequence being folded
Let $i = a$
Let $S = \{\}$
while $i \leq$ TRESHOLD($RNA$) do
    $S = S \cup$ COMPUTESLIDINGWINDOW($RNA$, $i$)
    $i = i \times b$
end while
return WEIGHTEDACTIVITYSELECTION($S$)
```

was found using empirical experimentation. In addition, a and b are arbitrary constants set by the user. I shall discuss how I found good values for them in Section 4.2.2. The `ComputeSlidingWindow` function was a modified implementation of Hofacker, Priwitzer, and Stadler's [15] algorithm, described in Section 5.1. It first computes all consecutive windows for a fixed size i, breaks these up into RNA intervals, and returns the set of these RNA intervals.

3.3.1 Time Complexity

The time complexity of this algorithm is non-trivial to deduce. The runtime of `ComputeSlidingWindow` is $O(nL^2)$ where L is the window size, and n is the length of the RNA. It is called once for every window size. In addition `WeightedActivitySelection` uses $O(n \log n)$ time, where n is the number of RNA intervals in the set S. I shall now show that *ab*-splat has a worst case time complexity of $O(n^2)$.

The sequence of window sizes explored by *ab*-splat given any values for a and b are as follows.

$$ab^0,\ ab^1,\ ab^2,\ ab^3\ \cdots ab^{\log_b \sqrt{n}-1} \tag{3.1}$$

This sequence is of course bounded by the `Treshold` function which means that all terms are $O(\sqrt{n})$. It follows that there are $O(\log_b \sqrt{n})$ terms in the series. Geometric progressions of this type have the following identity.

$$ab^0 + ab^1 + ab^2 + ab^3 + \cdots + ab^{n-1} = \frac{a(1-b^n)}{1-b} \tag{3.2}$$

Which means that the closed form of Equation 3.1 is as follows.

$$\frac{a(1 - b^{\log_b \sqrt{n}})}{1 - b} \tag{3.3}$$

Since a and b are both constant values, this is $O(\sqrt{n})$. Thus, the sum of all window sizes as defined in Equation 3.1 is $O(\sqrt{n})$. `ComputeSlidingWindow` is run for every window size defined in this sequence. Since the time complexity of `ComputeSlidingWindow` is $O(nL^2)$, and the sum of all window sizes is $O(\sqrt{n})$, the cumulative work is $O(n(\sqrt{n})^2)$ which trivially simplifies to $O(n^2)$. Now I shall show that this dominates the cost of running Weighted Activity Selection.

Every window size produces $O(n)$ windows of size L. Every such window can contain at most $O(L)$ RNA intervals. Because the sum of all windows sizes is $O(\sqrt{n})$, and we collect $O(n)$ sets of windows, we are left with an upper bound of $O(n^{\frac{3}{2}})$ RNA intervals in total.

Weighted Activity Selection uses $O(n \log n)$ time. As we have $O(n^{\frac{3}{2}})$ RNA intervals in the worst case, this leads to a worst case time complexity of $O(n^{\frac{3}{2}} \log n)$, which is dominated by $O(n^2)$. Computing the largest sliding window requires only $O(n)$ space, since it uses $O(L^2)$ space in all cases, and the largest window size is $O(\sqrt{n})$. Weighted Activity Selection additionally requires only $O(n^{\frac{3}{2}})$ space. As all of these RNA intervals must be stored in S, *ab*-splat will thus use $O(n^{\frac{3}{2}})$ memory in the worst case. In contrast, the Zuker algorithm requires $O(n^3)$ time and $O(n^2)$ space. The computational bottleneck is usually time, however.

CHAPTER 4

Method & Results

4.1 Materials

4.1.1 Environment

All algorithms were implemented and tested using Ubuntu 13.10 running on an Intel i5-3210m processor with four gigabytes of RAM. The GNU C Compiler version 4.8.2 was used to compile all C and C++ code.

4.1.2 Software

The ViennaRNA Package [22] was used as a base for all algorithms presented in this paper. This package contains many useful programs for working with RNA. The RNAfold and RNALfold modules were used in this investigation.

RNAfold is a modern implementation of Zuker's folding algorithm. It predicts the minimum free energy secondary structure of a RNA given the primary sequence. It can also calculate the Boltzmann partition function, producing a matrix of base pair probabilities. Additionally the RNAfold module can compute the energy of any arbitrary secondary structure, given a corresponding RNA primary sequence. The computed free energy is, of course, an approximation based on RNAfold's thermodynamic energy model.

The RNALfold module implements a sliding window RNA folding algorithm. It is designed to find all locally optimal secondary structures for a RNA of size n, using a window of fixed size L. RNALfold implements Hofacker, Priwitzer, and Stadler's algorithm [15], and thus uses $O(nL^2)$ time and $O(n + L^2)$ space. This is not the most optimal algorithm available. As discussed in Section 2.4, Horesh et al. [16] presented an algorithm that also folds consecutive windows using a similar model. However, they achieved a typical time complexity of $O(nL)$. This algorithm was not used because the implementation provided by the authors is based on an older energy model taken from the Mfold [47] package. RNALfold is based on the same energy model as RNAfold, which has been recently updated [22]. For the sake of accurate comparison to RNAfold, and to ensure a state of the art energy model, RNALfold was used.

Version 2.1.6 of the ViennaRNA package was used. The package was built from the C source code after minor modifications were made to the RNALfold module. The ViennaRNA makefile was used to compile the package. The makefile compiles numerous standalone console applications for ViennaRNA's modules. It also creates a static library called RNAlib. This library was linked at compile time, and used to call RNAfold and RNALfold in the novel algorithms implemented as part of this investigation. The `Lfold.c` and `Lfold.h` files (which RNALfold comprises) were modified so that, when the algorithm was executed, RNALfold returned a linked list of local secondary structures and their free energy. Before modification it would instead print them to the standard output stream. The modified versions of these files are available in the files associated with this report.

Statistical tests, regression analyses, and plots were generated using GNU Regression, Econometrics and Time-series Library [5] version 1.9.14.

4.1.3 Testing Set

The RNA secondary structures used to test algorithms presented in this paper were taken from the RNA STRAND database [3]. The RNA STRAND database is a free-to-use, curated collection of RNA secondary structures taken from various publicly available databases and publications. A subset of RNA structural data was extracted from the database. This subset contained only RNA structures that were marked having been verified using X-ray crystallography, or NMR imaging. It also comprises only whole RNAs; none of the RNAs used were fragments or subsequences of larger RNA molecules. Finally, no duplicates were allowed in the selected set. Hereafter, I shall refer to this collection of RNA secondary structures as the 'testing set'. The testing set contained 392 different RNA molecules ranging in length from 20 to 3032 nucleotides.

4.2 Test Configuration

4.2.1 RNA Interval Merging Algorithms

In order to test and compare the various RNA interval merging algorithms (outlined in Section 3.2), all the windows of size five to 500 were precomputed for every RNA in the testing set. This is to say that the sliding window algorithm was run for a window size of five, then six, up to and including size 500 for each RNA, and the resulting RNA intervals were cached. The minimum size of five was chosen because meaningful stems do not form with a size fewer than five nucleotides. The maximum of 500 was due to space and time constraints. Computing these windows took several days, and used 2.5 gigabytes of memory to store. Each selection algorithm was run using these data as input. Every possible window size was tried exhaustively. For every RNA, the single most accurate window size was recorded, along with the accuracy value. Accuracy was judged as the F-score, which is the harmonic mean of sensitivity and precision (defined in Section 2.3).

4.2.2 *ab*-splat

The aim of the first test done on *ab*-splat was to find good values for a and b. A brute force method was used to test a combinatorial set of values for a and b such that $10 \leq a \leq 30$ and $1.5 \leq b \leq 4.0$. In this method, all integer values for a were attempted, and values for b were generated with a step size of `0.1`. This method was exhaustive, but slow. To speed up computation, precomputed windows were used. This meant that the `Threshold` function was altered slightly to $\min(\sqrt{\texttt{RNA Length}} \times 9.5,\ 500)$, as only windows up to size 500 were precomputed. To compare different *ab* pairs, the testing set was randomly partitioned into two sets containing an equal number of RNAs. One of these sets I called the training set, the other I called the validation set. The aforementioned brute force search was done on the training set, and the F-scores for *ab* pairs recorded; *ab* pairs with highest F-scores were deemed the best. These scores were then validated by running the same procedure on the validation set, then attempting to correlate the scores of *ab* pairs between training and validation sets. This indicated if high scores in the training set were related to high scores in the validation set. Correlating scores also allowed me to determine if results found in the training set were statistically valid.

The motivation for the second test done on *ab*-splat was to compare it to RNAfold in terms of accuracy and run-time. The best *ab* pair was used to configure the *ab*-splat algorithm, which was then run on the full testing set, and its F-scores recorded. These F-scores were then compared directly to F-scores produced by RNAfold, which was also run on the full testing set. In addition to this, the time to execute each algorithm was recorded and compared. Precomputed windows were not used so that the runtime and accuracy could be fairly compared to RNAfold. Because of this, the standard `Threshold` function was used.

4.3 Test Procedure & Results

4.3.1 Interval Selection

The purpose of the following tests was to determine if any RNA interval merging algorithms (see Section 3.2) could potentially achieve better accuracy than globally optimizing algorithms. The RNAfold package is a modern implementation of the Zuker algorithm, which globally optimizes a thermodynamic scoring scheme. RNAfold is thus used for comparison. I predicted that some merging algorithms should have higher accuracy if local interactions are stronger than global interactions during RNA folding.

The average score between all RNA interval selection algorithms was compared (see Table 4.1). The highest scoring algorithm appeared to be Top-Down Selection, closely followed by Weighted Activity Selection, then by Bottom-Up Selection. To verify this performance gap, a Wilcoxon Signed-Rank test was done to compare the recorded F-scores for corresponding RNA. This test was chosen because the data recorded did not appear to be parametric. While this test reflected the small difference in averages between Weighted Activity Selection and Top-Down Selection ($z = 1.33066$, see Table 4.2), it was not statistically significant ($p > 0.05$). To further test Top-Down Selection and Weighted Activity Selection, another Wilcoxon Signed-Rank test was done on only RNAs of length ≥ 300 bases. This revealed a moderate but statistically significant difference, indicating that Weighted Activity Selection generally had higher F-scores for larger RNAs ($p < 0.001, z = 4.55591$). Finally, a Wilcoxon Signed-Rank test was done between Weighted Activity Selection and Score Selection to check for circular dominance. In accordance with the mean F-score values, Weighted Activity Selection appeared to have higher F-scores than Score Selection ($p < 0.001, z = 12.0957$).

A Wilcoxon Signed-Rank test was also used to compare the best recorded F-scores for all RNA interval selection algorithms, and those recorded for RNAfold (see Table 4.2). All tests were statistically significant ($p < 0.001$). All algorithms

Algorithm	Mean	Median
BUS	0.38122	0.35065
SS	0.68373	0.73098
WAS	0.70395	0.75709
TDS	0.71684	0.80000
RNAfold	0.57483	0.60870

Table 4.1: Summary statistics of recorded F-scores for Weighted Activity Selection (WAS), Top-Down Selection (TDS), Bottom-Up Selection (BUS), Score Selection (SS), and RNAfold.

Test Subjects	z-value	Two-tailed p-value
WAS & TDS	-1.33066	0.183301
WAS & TDS (RNA length $\geq$ 300)	4.55591	2.60801×10^{-6}
WAS & SS	5.94681	2.73418×10^{-9}
BUS & RNAfold	-9.13933	6.28392×10^{-20}
SS & RNAfold	12.0957	0
TDS & RNAfold	13.119	0
WAS & RNAfold	13.2082	0

Table 4.2: Results of Wilcoxon Signed-Rank testing for F-scores. Weighted Activity Selection (WAS), Top-Down Selection (TDS), Bottom-Up Selection (BUS), Score Selection (SS), and RNAfold are included.

but Bottom-Up Selection were shown to outperform RNAfold; Weighted Activity Selection in particular ($z = 13.2082$). Figure 4.1 clearly depicts this performance difference.

4.3.2 RNA Intervals For Prediction

The purpose of the following tests was to determine if the *ab*-splat algorithm (see Section 3.3) was more accurate than RNAfold. Additionally, further tests were undertaken to compare the runtime efficiency of both algorithms. It was expected that *ab*-splat would be more accurate than, and run faster than, RNAfold.

An Ordinary Least Squares linear regression was done to determine the correlation between F-scores recorded for *ab*-splat in the training and validation sets. Figure 4.2 depicts the resulting model. A strong correlation was found ($R^2 = 0.703$, $p < 0.001$) for scores in the training set versus scores in the validation set. The best *ab* value pair found in the training set was $a = 24$, $b = 1.8$. The *ab*-splat algorithm was configured using these values. As the data did not appear

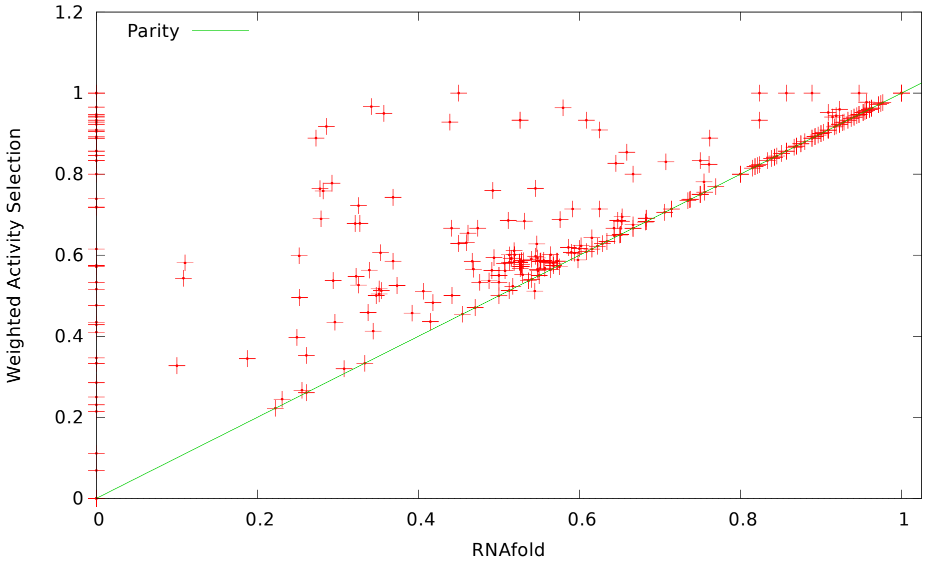

Figure 4.1: F-scores for matching RNA test cases achieved by Weighted Activity Selection and RNAfold plotted against one-another. The green line represents parity. Any points above the line are cases in which Weighted Activity Selection performed better; points below the line indicate that RNAfold was more accurate. Note that a maximum F-score is `1.0`.

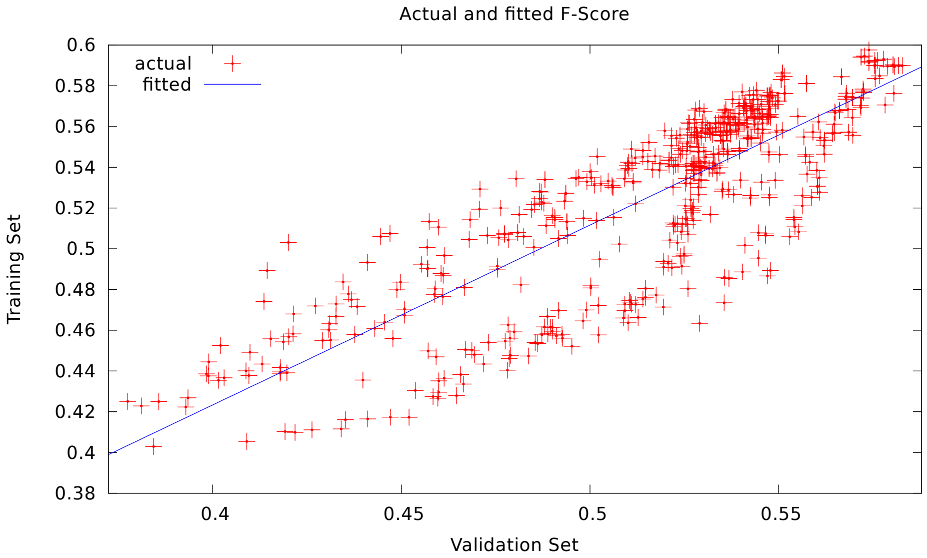

Figure 4.2: Correlation of F-scores for *ab*-splat in training set versus validation set. R-squared value = 0.702901, p-value < 0.0001. Standard Error of Regression = 0.025032.

to be normally distributed, a Wilcoxon Signed-Rank test was done to compare F-scores for *ab*-splat (with the best *ab* pair) and RNAfold. This test revealed a small difference in the F-scores in the population ($z = 0.1067$), which reflected the slight difference in averages (mean RNAfold score = 0.57483, *ab*-splat = 0.58588). However, this discrepancy was not statistically significant ($p > 0.05$).

Empirical observation was used to compare the runtime of RNAfold and *ab*-splat. A graph representation of the observed times is shown in Figure 4.3. This graph shows how the runtime of these algorithms grow with increasing RNA length. Both appeared to have a polynomial curve. Further analysis showed that both were best approximated by a polynomial regression. Both regression equations indicated a strong relationship between RNA length and algorithmic runtime ($R^2 > 0.99$). The best fit regression line for RNAfold runtime was $O(n^{2.374})$ where n is RNA length. The *ab*-splat algorithm has a best fit regression line that was $O(n^{2.25})$. A summary of the regression equations can be found in Table 4.3.

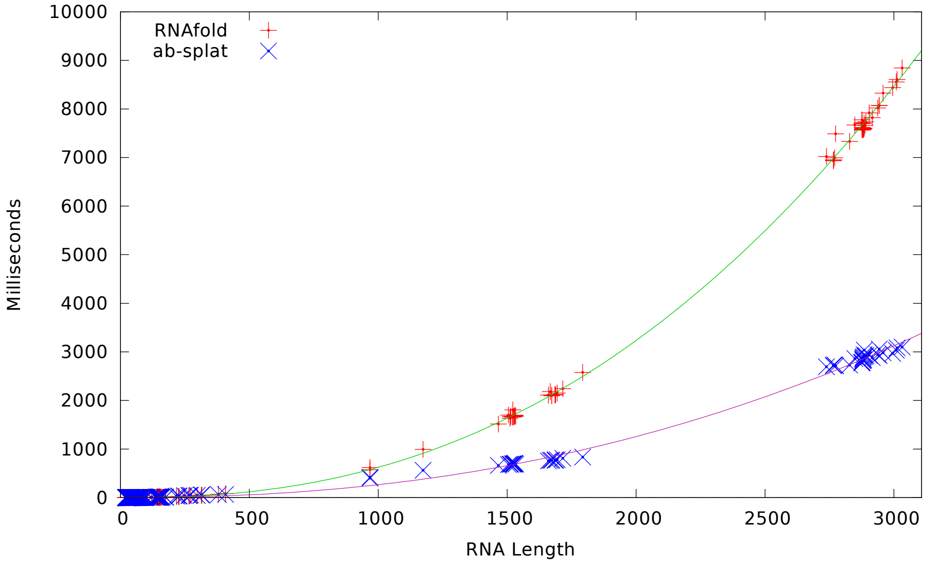

Figure 4.3: A comparison of the time taken for RNAfold to run against *ab*-splat. This graph indicates how much time RNAfold and *ab*-splat use with increasing RNA length. Best fit regression lines are also shown.

Algorithm	Regression Equation	R^2 value
RNAfold	$y = 0.358 + 4.7 \times 10^{-5} \times x^{2.374}$	0.9997
ab-splat	$y = 2.506 + 4.67 \times 10^{-5} \times x^{2.25}$	0.9985

Table 4.3: Summary of best fit regression lines for RNAfold and *ab*-splat.

CHAPTER 5

Discussion & Conclusions

5.1 Local Optimization

The results strongly indicate that there exists local windows that, when appropriately merged, are more accurate than globally optimal structures. Weighted Activity Selection, Top-Down Selection, and even Score Selection were markedly more accurate than RNAfold, having unequivocal statistical significance ($p \ll 0.001$). The central hypothesis of this investigation is that local interactions are stronger than global interactions during RNA folding. The Zuker algorithm, upon which RNAfold is based, finds the globally optimal secondary structure for a RNA. The RNA interval merging algorithms I presented built RNA secondary structures out of locally optimal sub-structures. It follows that local interactions must be stronger than global interactions for the thermodynamic model used in the ViennaRNA suite, as the locally optimizing algorithms performed much better than the globally optimizing algorithm.

Weighted Activity Selection in particular appeared to be the best RNA interval merging algorithm. Top-Down Selection had a slightly higher average, but the Wilcoxon Signed-Rank test indicated that this difference may be due to chance. In addition, Weighted Activity Selection performed better for larger RNAs (≥ 300 nucleotides), and this performance gap appeared to be statistically significant. Furthermore, as Weighted Activity Selection uses the free energy scores for RNA intervals, it should benefit from improved thermodynamic models.

Indeed, it is surprising that Top-Down Selection performed so well, as it ignores free energy entirely, instead choosing the largest possible RNA intervals. This may be because larger intervals will generally contain more bonds, and thus taking large intervals roughly approximates taking intervals with lower free energy. In addition, since the largest interval is taken at every step, one would expect that most of the RNAs' length would be covered by RNA intervals. Similarly, Score Selection, which greedily took the RNA intervals with highest score, had unexpectedly high accuracy. The explanation for this is more intuitive, as it greedily maximized the final score according to the thermodynamic model. Weighted Activity Selection seems to provide a good balance between these two approaches, seeking to find the combination of RNA intervals with maximum sum score. Bottom-Up Selection performed much worse than even RNAfold. I conjecture that this is because it does not optimize according to any useful scoring system; it always picks the smallest stems first, and ignores their free energy contribution.

It could be argued that the results only appear to show an improvement over RNAfold because the single best window size was recorded for each RNA in the testing set. To elucidate this claim further, I shall re-state the process used to compare my RNA interval merging algorithms. For each RNA in the testing set, and for each window size from five to 500, the RNA merging algorithm was run using the RNA intervals precomputed for the given window size. The window size with the single best F-score for a given RNA was recorded. If the thermodynamic hypothesis was correct, RNA structures should have the minimal free energy configuration. This means that no locally optimal window should have higher accuracy. Clearly this is not true for the free energy model used in ViennaRNA. It may then be argued that, since our current energy model is not perfect, the difference is due to chance. This is not necessarily true; many real RNAs fold into thermodynamically suboptimal states due to kinetic folding [12, 42]. Furthermore, my results show that all RNA interval selection algorithms (excluding Bottom-Up Selection) are consistently and significantly much more accurate across the entire test set. Statistically speaking, we can reject the hypothesis that this difference is due to chance. I suggest that the strength of local interactions in RNA folding needs to be given more weight in our current models if we are to improve their predictive power.

5.1.1 Future Work

The testing procedure used was incomplete, though compelling results were found nonetheless. Notably, only single window sizes were examined. As such, only RNA intervals found for a specific window size were used in a given secondary structure. It is possible that more accurate predictions could have been made using a combinatorial mix of window sizes to generate RNA intervals. This would provide the RNA interval merging algorithms with more intervals to choose from. Clearly this prodigiously increases the search space, which is why it was not attempted in this investigation. It might be fruitful for future research to assay combinatorial blends of many different window sizes. Additionally, while Weighted Activity Selection proved effective, there may exist better RNA interval merging algorithms.

Only the energy model implemented in RNAfold and RNALfold was used in this investigation. However, the findings presented here may also hold for other models of RNA folding. This is a classical argument from analogy. Nonetheless, locally optimized structures may prove more accurate for approaches based on SCFGs, or using machine learned parameters, or both. Because all such models are commonly based on assumptions inherited from the thermodynamic hypothesis, I argue that local optimization should work for any model of RNA secondary structure formation. The algorithms I have outlined are generic in that any folding algorithm could be used to generate sequential folds for a window of fixed size. It is important to note that using a Zuker-like energy model made this extremely efficient due to the existence of excellent sliding window algorithms; this is why it was the model of choice for this investigation.

5.1.2 Implications

I have found that there exists locally optimized secondary structures for RNAs which, in the average case, are at least 22% more accurate than the structures predicted by RNAfold. It follows directly that local interactions must be stronger than global interactions for many RNA molecules. In short, my hypothesis was strongly supported. However, it is important to consider the practical implications of this. Unfortunately, just because such a large accuracy reservoir exists, this does not mean it can be trivially used to improve existing algorithms, or to invent new algorithms. To achieve the full accuracy improvement, an oracle algorithm must somehow guess the correct window size to use for a RNA primary sequence. As I shall explain in the following section, this does not appear to be an easy task. Regardless, it should be possible to leverage this knowledge to improve existing algorithms, or to create new algorithms with superior accuracy.

Earlier (in Section 2.8), I extracted a single, salient finding from the work of Rivas [34]: that all RNA prediction algorithms seem to hit an upper limit in accuracy. I also suggested an explanation for this: that, for various historical reasons, they all implicitly adhere to the Thermodynamic Hypothesis as described by Anfinsen [4]. I submit that the findings presented here suggest a way beyond the Thermodynamic Hypothesis, and as a result a way past the accuracy bound observed for modern RNA folding techniques.

5.2 The *ab*-splat Algorithm

Having found support for my hypothesis, I aimed to improve the accuracy and speed of RNA secondary structure prediction by using locally optimal windows. This aim was only partially met, but the resulting algorithm still has practical applications nonetheless. This algorithm, which I have called *ab*-splat, is theoretically faster than any implementation of the Zuker algorithm. Furthermore, the Zuker algorithm is the fastest known RNA prediction algorithm which has reasonable predictive accuracy. As I shall now argue, *ab*-splat has at least comparable accuracy to RNAfold, which is a state of the art implementation of the Zuker algorithm, and is also faster both in theory, and in practice.

5.2.1 Accuracy

My results (details can be found in Section 4.3.2) showed that *ab*-splat was slightly more accurate than RNAfold, as it had a small (roughly 2%) advantage in F-score. However, further testing revealed that this difference was not statistically significant. The null hypothesis for the test used (the Wilcoxon Signed-Rank test) states that there is no difference between the two populations tested. Because this hypothesis could not be rejected ($p > 0.05$), I conclude that the *ab*-splat algorithm has accuracy comparable to RNAfold, as there was no statistically significant difference between the F-scores recorded for both algorithms. This may be a limitation of my testing set. Though it was composed of 392 RNAs, it is possible that *ab*-splat could have better or worse relative performance on other data sets. More research must be done before one can definitely say that either algorithm has greater predictive power than the other.

5.2.2 Computational Complexity

I have shown that the time complexity of the *ab*-splat algorithm is $O(n^2)$ in the worst case, given a RNA of length n. The Zuker algorithm (and thus RNAfold) requires $O(n^3)$ time. I have also shown that this speed-up is tangible in practice through empirical testing. This is of practical importance, as RNA folding can often take considerable computational time. Additionally, *ab*-splat can be made to run efficiently on parallel architectures. Instead of using a sliding window repeatedly with increasing sizes, all windows could be computed in parallel. Thence one could also optimize Weighted Activity Selection by using a parallel sorting algorithm, and computing the q array in parallel. Unfortunately, the parallel version was not implemented; it is therefore a viable direction for future research.

Memory usage was a less important criteria when analysing the performance of *ab*-splat, as time is usually the bottleneck during RNA secondary structure prediction. As a result, I did not test this aspect of *ab*-splat's performance. However, I have shown that the theoretical worst case space complexity is $O(n^{\frac{3}{2}})$; this is more frugal than the Zuker algorithm, which requires $O(n^2)$ space.

5.2.3 A Better Algorithm

The notable shortcoming of the *ab*-splat algorithm is that it is, at best, not much more accurate than current algorithms. In this sense, the aims of my investigation were not met. This deficiency is unequivocal, and perhaps a little unexpected, given that my original hypothesis was strongly supported, and there do exist locally optimal windows that, when merged, are much more accurate than a globally optimal solution. I now discuss various avenues of inquiry which I believe will lead to algorithms capable of leveraging the reservoir of accuracy *ab*-splat was not able to utilize.

The landscape of F-scores (as a function of window size) appeared extremely rugged, often with local minima and maxima in close proximity. This implies that choosing good window sizes is difficult. Indeed, I attempted to define a function that, given a primary RNA sequence, would predict a good window size. I had no success. Features such as the mean and median stem size found by RNAfold did not appear to correlate well with good window sizes. Intriguingly, the length of the RNA did correlate very roughly with accurate window sizes. This proved difficult to use in practice due to the extreme ruggedness of the accuracy landscape. It is possible that machine learning algorithms could find and use features of the primary sequence to make reliable guesses about good window sizes.

Features used for RNA design are viable candidates. RNA design is the reverse problem to RNA prediction: given a target secondary structure, we must find a primary sequence that is most likely to fold into it. Because of the computational difficulty of the problem, finding reliable heuristics or rules is useful. Lee et al. [20] had tens of thousands of online, human participants learn to design RNA molecules. Using the insights found by these participants, and information found using machine learning, they confirmed several previously postulated rules for RNA design, and found many new ones. Some of these should be readily applicable here. For example, they confirmed that G-C base pairs usually close multiloops, and that adenine concentration is unusually high outside of stems. Furthermore, they found useful rules for guanine base placement at the end of hairpin loops. These are just some of the readily applicable rules that an 'oracle' type algorithm could use to predict good window sizes using only the RNA primary sequence. A hypothesis about various stems could be formed, and thence some prediction about the best window size.

Though I have not incorporated pseudoknots into the algorithms presented, this does not mean that they should be overlooked. An improved algorithm could encompass pseudoknot prediction. Because the *ab*-splat algorithm (and any similar algorithm) builds RNA structures out of optimally folded smaller structures, pseudoknots could be added as a subsequent step. In contrast, likely pseudoknot stems could be found in a preprocessing step, then a suitable window size could be inferred using these stems.

5.3 Conclusions

I supposed that local interactions might be stronger than global interactions during RNA folding. My hypothesis was based on the ruggedness of the energy landscape, the evidence for kinetic folding, and upon the accuracy ceiling of modern prediction algorithms. If this supposition were true, one might expect to find that more accurate structures could be predicted using only narrow windows of optimization. This is precisely what I have found. There exist combinations of locally optimal RNA segments that, when combined, are more accurate than a globally optimal RNA secondary structure prediction. These can be found reliably for most RNA molecules and are often much more accurate than their globally optimal counterparts. In an attempt to leverage these findings, I created the *ab*-splat algorithm. The *ab*-splat algorithm was based on the idea of finding and merging RNA structures generated using windows of exponentially increasing size. Though crude, this approach was surprisingly effective. Interestingly, it was effective in an unexpected way. RNAfold was shown to have comparable accuracy to *ab*-splat. However, the space and time requirements of *ab*-splat are much better than the Zuker algorithm. This makes *ab*-splat eminently practical when speed is required.

These findings run counter to the Thermodynamic Hypothesis, which posits that biologically active molecules form structures with minimum Gibbs free energy. Presented in this report are predicted structures that do not have minimal free energy under a state of the art thermodynamic model. Despite this, they are more accurate than structures that do. Clearly the current model is insufficient to explain how RNAs fold. While future improvements to this model may encompass local interactions, for now it appears that we should look beyond the Thermodynamic Hypothesis.

Bibliography

[1] Alberts, B., Bray, D., Hopkin, K., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. *Essential Cell Biology*, third ed. Garland Science: New York, 2009.

[2] Amaral, P. P., Dinger, M. E., Mercer, T. R., and Mattick, J. S. The eukaryotic genome as an rna machine. *Science 319*, 5871 (2008), 1787–1789.

[3] Andronescu, M., Bereg, V., Hoos, H. H., and Condon, A. Rna strand: the rna secondary structure and statistical analysis database. *BMC Bioinformatics 9*, 1 (2008), 340.

[4] Anfinsen, C. Principles that govern the protein folding chains. *Science 181* (1973), 233–230.

[5] Baiocchi, G., and Distaso, W. Gretl: Econometric software for the gnu generation. *Journal of Applied Econometrics 18*, 1 (2003), 105–110.

[6] Beaton, M. J., and Cavalier-Smith, T. Eukaryotic non-coding dna is functional: evidence from the differential scaling of cryptomonad genomes. *Proceedings of the Royal Society of London. Series B: Biological Sciences 266*, 1433 (1999), 2053–2059.

[7] Bellaousov, S., Reuter, J. S., Seetin, M. G., and Mathews, D. H. Rnastructure: web servers for rna secondary structure prediction and analysis. *Nucleic Acids Research 41*, W1 (2013), W471–W474.

[8] Condon, A. Problems on rna secondary structure prediction and design. In *Automata, Languages and Programming*. Springer, 2003, pp. 22–32.

[9] Conn, G. L., and Draper, D. E. Rna structure. *Current Opinion in Structural Biology 8*, 3 (1998), 278–285.

[10] Dawson, W., Takai, T., Ito, N., Shimizu, K., and Kawai, G. A new entropy model for rna: part iii, is the folding free energy landscape of rna funnel shaped? *Journal of Nucleic Acids Investigation 4*, 1 (2013).

[11] Deogun, J. S., Donis, R., Komina, O., and Ma, F. Rna secondary structure prediction with simple pseudoknots. In *Proceedings of the second conference on Asia-Pacific bioinformatics-Volume 29* (2004), Australian Computer Society, Inc., pp. 239–246.

[12] Ditzler, M. A., Rueda, D., Mo, J., Håkansson, K., and Walter, N. G. A rugged free energy landscape separates multiple functional rna folds throughout denaturation. *Nucleic Acids Research 36*, 22 (2008), 7088–7099.

[13] Do, C. B., Woods, D. A., and Batzoglou, S. Contrafold: Rna secondary structure prediction without physics-based models. *Bioinformatics 22*, 14 (2006), e90–e98.

[14] Hofacker, I. L. Rna consensus structure prediction with rnaalifold. In *Comparative Genomics*. Springer, 2008, pp. 527–543.

[15] Hofacker, I. L., Priwitzer, B., and Stadler, P. F. Prediction of locally stable rna secondary structures for genome-wide surveys. *Bioinformatics 20*, 2 (2004), 186–190.

[16] Horesh, Y., Wexler, Y., Lebenthal, I., Ziv-Ukelson, M., and Unger, R. Rnaslider: a faster engine for consecutive windows folding and its application to the analysis of genomic folding asymmetry. *BMC Bioinformatics 10*, 1 (2009), 76.

[17] Kato, Y., Seki, H., and Kasami, T. Stochastic multiple context-free grammar for rna pseudoknot modeling. In *Proceedings of the Eighth International Workshop on Tree Adjoining Grammar and Related Formalisms* (2006), Association for Computational Linguistics, pp. 57–64.

[18] Koessler, D. R., Knisley, D. J., Knisley, J., and Haynes, T. A predictive model for secondary rna structure using graph theory and a neural network. *BMC Bioinformatics 11*, Suppl 6 (2010), S21.

[19] Kramer, F. R., and Mills, D. R. Secondary structure formation during rna synthesis. *Nucleic Acids Research 9*, 19 (1981), 5109–5124.

[20] Lee, J., Kladwang, W., Lee, M., Cantu, D., Azizyan, M., Kim, H., Limpaecher, A., Yoon, S., Treuille, A., and Das, R. Rna design rules from a massive open laboratory. *Proceedings of the National Academy of Sciences* (2014), 201313039.

[21] Leung, Y. Y., Ryvkin, P., Ungar, L. H., Gregory, B. D., and Wang, L.-S. Coral: predicting non-coding rnas from small rna-sequencing data. *Nucleic Acids Research 41*, 14 (2013), e137–e137.

[22] Lorenz, R., Bernhart, S. H., Zu Siederdissen, C. H., Tafer, H., Flamm, C., Stadler, P. F., Hofacker, I. L., et al. Viennarna package 2.0. *Algorithms for Molecular Biology 6*, 1 (2011), 26.

[23] Lyngsø, R. B., and Pedersen, C. N. Rna pseudoknot prediction in energy-based models. *Journal of Computational Biology 7*, 3-4 (2000), 409–427.

[24] Makeyev, E. V., and Maniatis, T. Multilevel regulation of gene expression by micrornas. *Science 319*, 5871 (2008), 1789–1790.

[25] Mattick, J. S. A new paradigm for developmental biology. *Journal of Experimental Biology 210*, 9 (2007), 1526–1547.

[26] Morgan, S. R., and Higgs, P. G. Evidence for kinetic effects in the folding of large rna molecules. *The Journal of Chemical Physics 105*, 16 (1996), 7152–7157.

[27] Namy, O., Moran, S. J., Stuart, D. I., Gilbert, R. J., and Brierley, I. A mechanical explanation of rna pseudoknot function in programmed ribosomal frameshifting. *Nature 441*, 7090 (2006), 244–247.

[28] Neidle, S. *Principles of Nucleic Acid Structure*. Academic Press, 2010.

[29] Nussinov, R., and Jacobson, A. B. Fast algorithm for predicting the secondary structure of single-stranded rna. *Proceedings of the National Academy of Sciences 77*, 11 (1980), 6309–6313.

[30] Nussinov, R., Pieczenik, G., Griggs, J. R., and Kleitman, D. J. Algorithms for loop matchings. *SIAM Journal on Applied mathematics 35*, 1 (1978), 68–82.

[31] Pontius, J. U., Mullikin, J. C., Smith, D. R., Lindblad-Toh, K., Gnerre, S., Clamp, M., Chang, J., Stephens, R., Neelam, B., Volfovsky, N., et al. Initial sequence and comparative analysis of the cat genome. *Genome Research 17*, 11 (2007), 1675–1689.

[32] Proctor, J. R., and Meyer, I. M. Cofold: an rna secondary structure prediction method that takes co-transcriptional folding into account. *Nucleic Acids Research 41*, 9 (2013), e102–e102.

[33] Reeder, J., and Giegerich, R. Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC Bioinformatics 5*, 1 (2004), 104.

[34] RIVAS, E. The four ingredients of single-sequence rna secondary structure prediction. a unifying perspective. *RNA Biology 10*, 7 (2013), 1185.

[35] RIVAS, E., AND EDDY, S. R. A dynamic programming algorithm for rna structure prediction including pseudoknots. *Journal of Molecular Biology 285*, 5 (1999), 2053–2068.

[36] RIVAS, E., LANG, R., AND EDDY, S. R. A range of complex probabilistic models for rna secondary structure prediction that includes the nearest-neighbor model and more. *RNA 18*, 2 (2012), 193–212.

[37] SPERSCHNEIDER, J., AND DATTA, A. Dotknot: pseudoknot prediction using the probability dot plot under a refined energy model. *Nucleic Acids Research 38*, 7 (2010), e103–e103.

[38] SPERSCHNEIDER, J., DATTA, A., AND WISE, M. J. Heuristic rna pseudoknot prediction including intramolecular kissing hairpins. *RNA 17*, 1 (2011), 27–38.

[39] STUDNICKA, G. M., RAHN, G. M., CUMMINGS, I. W., AND SALSER, W. A. Computer method for predicting the secondary structure of single-stranded rna. *Nucleic Acids Research 5*, 9 (1978), 3365–3388.

[40] TAUFER, M., LICON, A., ARAIZA, R., MIRELES, D., VAN BATENBURG, F., GULTYAEV, A. P., AND LEUNG, M.-Y. Pseudobase++: an extension of pseudobase for easy searching, formatting and visualization of pseudoknots. *Nucleic Acids Research 37*, suppl 1 (2009), D127–D135.

[41] TINOCO JR, I., AND BUSTAMANTE, C. How rna folds. *Journal of Molecular Biology 293*, 2 (1999), 271–281.

[42] TREIBER, D. K., AND WILLIAMSON, J. R. Beyond kinetic traps in rna folding. *Current Opinion in Structural Biology 11*, 3 (2001), 309–314.

[43] VAN BATENBURG, F., GULTYAEV, A. P., AND PLEIJ, C. W. An apl-programmed genetic algorithm for the prediction of rna secondary structure. *Journal of Theoretical Biology 174*, 3 (1995), 269–280.

[44] WATSON, J. D., CRICK, F. H., ET AL. Molecular structure of nucleic acids. *Nature 171*, 4356 (1953), 737–738.

[45] WIESE, K. C., DESCHENES, A. A., AND HENDRIKS, A. G. Rnapredictan evolutionary algorithm for rna secondary structure prediction. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on 5*, 1 (2008), 25–41.

[46] Xu, Z., Almudevar, A., and Mathews, D. H. Statistical evaluation of improvement in rna secondary structure prediction. *Nucleic Acids Research 40*, 4 (2012), e26–e26.

[47] Zuker, M. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research 31*, 13 (2003), 3406–3415.

[48] Zuker, M., and Stiegler, P. Optimal computer folding of large rna sequences using thermodynamics and auxiliary information. *Nucleic Acids Research 9*, 1 (1981), 133–148.