A GENETIC ALGORITHM FOR RNA SECONDARY STRUCTURE PREDICTION USING STACKING ENERGY THERMODYNAMIC MODELS

by

Alain Deschênes

B. Sc., Chemistry (Honours), University of New Brunswick, 2002

B. C. S., University of New Brunswick, 2002

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Interactive Arts and Technology

SIMON FRASER UNIVERSITY
Summer 2005

APPROVAL

Name: Alain Deschênes

Degree: Master of Science

Title of thesis: A Genetic Algorithm for RNA Secondary Structure Prediction using Stacking Energy Thermodynamic Models

Examining Committee:

Dr. Rob Woodbury, Professor, Information Technology
Simon Fraser University

Dr. Kay C. Wiese, Assistant Professor, Computing Science
Simon Fraser University
Senior Supervisor

Dr. Vive Kumar, Assistant Professor, Information Technology
Simon Fraser University
Supervisor

Dr. Belgacem Ben Youssef, External Examiner
Assistant Professor, Information Technology
Simon Fraser University

Date Approved: April 18, 2005

SIMON FRASER UNIVERSITY

PARTIAL COPYRIGHT LICENCE

Abstract

RNA structure is an important field of study. Predicting structure can overcome many of the issues with physical structure determination.

Structure prediction can be simplified as an energy minimization problem. Common optimization techniques are the DPA and the GA.

RnaPredict is a GA used for RNA secondary structure prediction using energy minimization and is evolved from Dr. Wiese's lab. Selection, recombination, mutation, and elitism are used to optimize the candidate structures in a population. Candidate solutions get closer to the global energy optimum with each generation.

This thesis focuses on the addition of a hydrogen bond model and two stacking energy models, and studies their relative merits. It also studies different types of encoding used in the GA.

The prediction accuracy is compared with known structures, the Nussinov DPA predictions and the *mfold* DPA predictions. RnaPredict is able to predict more accurate structures than Nussinov and performs similarly to *mfold*.

To my parents...

"Go west, young man"

— John B. L. Soule, TERRE HAUTE *Express, 1851*

Acknowledgments

There are numerous people and organizations that have supported me during the course of this research. Without their support, I would not have been able to complete this thesis and I feel they should be acknowledged.

I would first like to thank Dr. Kay C. Wiese for being my senior supervisor. Dr. Wiese has been available throughout the course of my studies at Simon Fraser University. He helped me find scholarship opportunities to ensure I would have a steady source of income.

Dr. Wiese also helped me choose a topic for my thesis through a research assistantship and directed reading classes. He guided my work so it would be directly applicable to my thesis.

Dr. Wiese also encouraged me to publish work that I had done as a research assistant and also sponsored travel for presentations at numerous international conferences.

Any work I would ever submit to him was returned during a scheduled meeting where he would provide constructive criticism helping me improve its quality. I found this to be highly useful and rewarding. In the end, Dr. Wiese has been, not only an excellent supervisor, but also, a great friend.

I would also like to acknowledge organizations that provided me with numerous scholarships and grants. I would especially like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for awarding me a Post Graduate Scholarship (PGS-A) providing for 20 months of funding. I should also mention other organizations who provided me with scholarships, grants, and awards. These include the Congress on Evolutionary Computation (CEC 2003), the Faculty of Applied Sciences at Simon Fraser University (SFU), the School of Interactive Arts and Technology (SIAT) at SFU, Institute of Electrical and Electronics Engineers (IEEE), the Canadian Conference on Artificial Intelligence, the Symposium on Computational Intelligence in Bioinformatics and Computational

Biology (CIBCB), and the Advanced Systems Institute of BC (ASI).

I would like to thank Dr. Rob Woodbury and his committee for their hard work and dedication to get the new graduate program approved. Even with numerous hurdles, they never got discouraged. The approval of the program gave me the opportunity to write this thesis and for that, I am grateful.

I would like to thank my colleagues in Dr. Wiese's research group. These include Edward Glen, Andrew Hendriks, Kirt Noël, Jagdeep Poonian, and Herbert Tsang. They were instrumental in all aspects of my research including everything from helping solve technical computing issues to discussing details of RNA folding. In particular, Edward Glen and Andrew Hendriks designed and implemented large sections of the current incarnation of RnaPredict. Edward Glen also created a visualization software, jViz.Rna, which was used to create structural RNA diagrams. Jagdeep Poonian contributed to this work by providing an implementation of the Nussinov DPA. Andrew Hendriks also allowed me to use some presentation material as the basis for some of my presentations. Lastly, Kirt Noël answered all sorts of questions on the subject of molecular biology. He also entertained me over the years with his endless wit and sarcasm.

For technical support, I would like to thank three people in particular. These are Gordon Pritchard, Patrick Lougheed, and Robin Johnson. With their help, I had access to a high performance computing centre that they administered. They were very accommodating in making sure that the hardware and software I required would be available at all times. They also never shied away from the challenges of creating complex custom solutions to accommodate my needs.

I would also like to thank Allison Neil, Heather Clendenning, and Joyce Black for helping me with numerous requests during my studies.

I am grateful to Dr. Vive Kumar for accepting to be on my supervisory committee, and also to Dr. Belgacem Ben Youssef for accepting to be my external examiner.

I would like to offer my gratitude to both Alexandre Deschênes and Marie-Claude Lavoie for proofreading my thesis on an extremely tight schedule. Their numerous corrections improved the quality of this document tremendously.

Lastly, I would like to offer my most sincere gratitude to my parents, Alexandre and Huguette Deschênes. From the beginning of my studies, they have supported me through every decision, while sometime not understanding the reasons. They have also been a source of financial support making my life in the Vancouver area much more comfortable

and entertaining. Without their help, I would definitely not be where I am today.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The field of computational RNA secondary structure prediction has a short but active history. The first important work was done in the field of dynamic programming algorithms (DPAs). The pioneering work in this domain was done by Nussinov in 1978 [4] with the introduction of her algorithm used to maximize the number of base pairs in a structure. Zuker [5, 6, 7, 8, 9, 10] later introduced another DPA that optimizes the free energy of structures using a thermodynamic model. The development of *mfold* is still active today [10] and has become the benchmark for computational RNA secondary structure prediction. A variant of *mfold* is RNAStructure written by Mathews [11]. This variant ports *mfold* from C for Unix to C++ for the Microsoft Windows platform, adding a graphical user interface (GUI). Another DPA for secondary structure prediction was developed by Hofacker [12]. This package is called Vienna and includes three kinds of DPAs. The first gives a single structure with optimal free energy, similar to Zuker's *mfold*. The second calculates the base pair probabilities in the thermodynamic ensemble using a partition function [13, 14], and the third can generate all suboptimal structures within a given energy range of the optimal energy [15, 16]. Nussinov and *mfold* will be discussed in more detail in Chapters 8 and 9, respectively.

Kinetic folding has also had some success in structure prediction. Kinetic folding simulates stochastic folding of RNA sequences. The algorithm models the formation, dissociation, and shifting of individual base pairs. KinFold is a software implementation from Flamm et al. [17, 18].

The class of methods yielding the highest degree of accuracy in predicting RNA secondary structures is comparative methods. These methods make use of more than one phylogenetically related sequences and try to find folding patterns between them [19, 20]. Current implementations based on the Sankoff algorithm are Foldalign [21, 22], Dynalign [23], and PMcomp [24]. One implementation that combines energy minimization and comparative sequence analysis is Dynalign. PMcomp includes sequence alignment with maximal pairing. A comprehensive review of comparative RNA structure prediction approaches can be found in [25].

More recently, population based approaches have been developed. Evolutionary algorithms (EAs) have been used since the mid-1990s in this domain. The early successful applications of EAs were those of van Batenburg [26, 27], Shapiro [28], and Benedetti [29]. Early developments include simulating folding pathways [30] via genetic algorithms (GAs), implementation of a massively parallel version [31, 32], and the introduction of an annealing mutation operator [33]. This thesis builds on this research and the research conducted in Dr. Wiese's lab. These developments include a permutation based GA, called RnaPredict [1], studies of selection, crossover operators, and representation issues [34, 35, 36]. A study of thermodynamic models was also done [37]. RnaPredict was also parallelized [38, 39] improving on the serial version. GAs will be discussed in more detail in Chapter 5.

1.1 Research question

The research question for this project is the secondary structure prediction of RNA molecules using only the primary sequence as input. The research incorporates three new thermodynamic models into RnaPredict [34] and demonstrates the high prediction accuracies due to the improved models.

There are numerous objectives in this thesis. The first goal is to further establish the benefits of using permutation encoding over binary encoding in the domain of RNA secondary structure prediction. The relative merits of the different binary and permutation crossover operators are discussed. Also, the relative merits of two selection strategies, Standard Selection (STDS) and Keep-Best Reproduction (KBR), are investigated. Various different crossover and mutation rate combinations were tested to find optimal settings.

Different thermodynamic models were implemented and tested. One additional hydrogen bond model as well as two stacking energy models were integrated into RnaPredict. The

relative merit of these four different thermodynamic models was determined.

RnaPredict generates candidate low energy structures for comparison to the natural fold. The accuracy of these predictions is investigated along with a comparison to other prediction methods such as the Nussinov DPA and the *mfold* DPA.

1.2 Thesis breakdown

This thesis is divided in multiple chapters. RNA will be discussed in further detail in Chapter 2. Chapter 3 will detail the four thermodynamic models included in RnaPredict while Chapter 4 shows how this energy relates to structure prediction accuracy. Chapter 5 introduces RnaPredict, a GA for RNA secondary structure prediction. Chapter 6 describes how the GA parameters were optimized. Chapter 7 compares the structures predicted by RnaPredict to known structures, while Chapters 8 and 9 compare the predicted structures to those generated by the Nussinov DPA and the *mfold* DPA.

Chapter 2

Ribonucleic acid

RNA [40] is an important biological molecule, that is, one of the building blocks in living cells. Simply, RNA is a linear polymer of ribose sugar rings linked together by phosphate groups. Each one of these sugar rings has one of four different basic nitrogenous functional groups attached. These four groups are adenine, cytosine, guanine, and uracil (A, C, G, and U). The polymer formed is composed of a particular sequence of these four building blocks.

2.1 Nucleotides

Upon closer inspection, a single nucleotide can be broken down into two parts. There is a part, common to all nucleotides, which is the sugar-phosphate backbone; the other part is called the base. A nucleotide without a phosphate group is called a nucleoside.

The base and ribose sugars of the nucleotides are heterocyclic compounds. These compounds are formed of carbon, nitrogen, oxygen, and hydrogen atoms.

The bases' structure themselves have another classification. Adenine and guanine are bases that contain two rings in their structure and are called purines. The other two bases contain only one ring in their structure and are called pyridines.

2.2 RNA strands

Nucleotide monomer units link together to form a polymer. The atoms on the nucleotides are numbered following standard chemical conventions. The linking is made between the oxygen on the 5′-phosphate and the 3′-hydroxyl on the ribose sugar, where the prime (′) is

used to distinguish the numbering from the atoms of the nitrogenous base. These covalent bonds are formed through a phosphodiester linkage. The polymer formed is a long chain that can be either less than fifty nucleotides, called oligonucleotide, or counting up to many thousands of nucleotides, called polynucleotides. The length is affected by the class and functionality of the RNA strand. Strands also have a directionality. During their synthesis, elongation proceeds in the $5' \rightarrow 3'$ direction at a rate of 50-100 bases per second. By definition, the $5'$ end lacks a nucleotide at the $5'$ position and the $3'$ end lacks a nucleotide at the $3'$ position.

2.3 Function of RNA

RNA molecules serve a key role in the translation of the information encoded in DNA in the synthesis of protein. RNA has three main functions that are involved in protein synthesis. First, the genes encoded by the DNA are copied to messenger RNA (mRNA) in the cell nucleus in a process called transcription. These mRNA strands are sent to the cell ribosome acting as messengers from the nucleus. The ribosomes are composed of ribosomal RNA (rRNA) and protein molecules. In the ribosome, the mRNA is read and amino acids sequences are assembled to form protein material according to the information on the mRNA strand. When protein synthesis occurs at the ribosome, the amino acids are found to be bonded to a molecule of transfer RNA (tRNA) which forms a complex shape so that it bonds to a specific amino acid and has an exposed anticodon (reverse of the RNA triplet for that amino acid). In the ribosome, each mRNA triplet is matched to the opposite tRNA anticodon, one triplet at a time as each amino acid bonds to the one before it, creating a polypeptide.

The genetic code maps sequences of three nucleotide bases, called codon, into amino acids that form the building blocks of protein polymer chains. Some important considerations relating to encoding RNA into amino acids are directionality, the size of a codon, relationship between codons, and the reading frame. First, RNA strands have a rigid mapping to amino acids. The $5'$ of RNA is mapped to the amino end of polypeptide while the $3'$ end corresponds to the carboxy end. The hypothesis that triplets were the basis of coding for amino acid came from the idea that three nucleotide sequences could potentially code for 64 different amino acids while 20+ are now known.

A particular nucleotide can only be part of a single codon, that is, the sequence is read

three nucleotides at a time. Once a triplet is read, the reading window is shifted ahead by a full triplet. A direct consequence of this is that single nucleotide mutation cannot affect more than one amino acid. Lastly, proteins are encoded and their chain lengths are controlled by very specific codons. In the complete set of available codons, a few are reserved for starting (start-codons) and stopping (stop-codons) the amino acid sequence.

Other than its involvement with protein synthesis, RNA have been found to act as a catalyst for some biochemical reactions [41]. An early review article [42] describes some of these catalyst roles:

> "A number of RNA enzymes (ribozymes) are known to exist in nature, and these serve as a starting point from which to begin an evolutionary search for novel catalysts. It has been possible, for example, to convert an RNA enzyme that cleaves single-stranded RNA to an RNA enzyme that cleaves single-stranded DNA. It has also been possible to evolve RNA metalloenzymes that have novel metal dependence. It remains to be seen to what extent the range of RNA-based catalytic function can be expanded. If nature provides any indication, it is that the catalytic prowess of RNA is rather limited. After all, proteins carry out most of the catalytic functions in biological organisms. RNA has been shown to catalyze phosphoester transfer reactions, phosphoester hydrolysis, aminoacyl ester hydrolysis and peptide bond formation. Considering the functional groups that exist within RNA and the ability of RNA to adopt a well-defined tertiary structure, a member of other catalytic functions seem feasible. Nonetheless, proteins are more versatile catalysts, containing twenty dissimilar amino acid components rather than the four similar nucleotide components of RNA."

A second review can be found in [43]. The review discusses how RNase P is involved in tRNA maturation and that self-splicing introns are involved in mRNA maturation.

2.4 RNA structure

In the cell, RNA usually exists as a single strand. A strand is fairly flexible and tends to fold back onto itself where intra-molecular hydrogen bonds can form between certain base pairs [41]. The strongest interactions form between pairs that are characterized as complementary. In RNA, adenine is complementary to uracil where two hydrogen bonds

can form between them and cytosine is complementary to guanine where three hydrogen bonds can form. These types of pairs are called Watson-Crick pairs. Another important pair can form between guanine and uracil called a GU wobble pair containing two hydrogen bonds. Watson-Crick and GU pairs form a group called canonical base pairs. Other weaker interactions are allowed between many different combinations of base pairs.

2.4.1 Primary structure

RNA structure has three representations. Each of these is used for different abstractions of the true structure. First, the primary structure describes the sequence and is written as a textual string using the letters A, C, G, and U to denote adenine, cytosine, guanine and uracil. By convention, the string is always written with the $5'$ end at the left to the $3'$ end at the right, that is, in the $5' \rightarrow 3'$ direction. This type of representation only describes the sequence. Separating sequence from structure can be useful when looking for patterns in the sequence.

A common type of experiment on sequences is called a sequence alignment. Briefly, sequence alignment concerns itself with the relationship between RNA sequences. The idea is to try to align different, but related sequences, by adding and removing gaps. The purpose is to correlate sequence and function across genomes.

2.4.2 Secondary structure

When an RNA strand folds onto itself, it forms hydrogen bonds between certain base pairs. Each nucleotide usually has the possibility of being paired with a maximum of one other nucleotide by base pairing. In this process intra-molecular hydrogen bonds form between certain bases. The most stable pairs form between GC, AU, and GU and their mirrors, CG, UA, and UG. Their stability also makes them the most common pairs. These pairs are called canonical base pairs. The collective listing of the paired bases in an RNA molecule is what is called the secondary structure.

Using base pairing rules allow for enumeration of all the pairs that have the potential to form. Due to the combinatorial nature of the problem, the challenge is to predict which ones will form to yield the natural fold.

Another property of secondary structure is that pairs tend to form in groups yielding higher order structure. Common RNA substructures are hairpin loops, internal loops and

bulges, multi-branch loops, and dangling ends. Forming adjacent pairs tends to increase the stability. These stacked pairs are also called stems, or helices. Formally, stacked pairs exist when two or more base pairs

$$(i,j),\ldots,(i+n,j-n), 1 \leq n < m, \text{ where } m = \frac{j-i-3}{2} \tag{2.1}$$

exist such that the ends of the pairs are adjacent, forming a helical structure. In this equation, n is an integer taking values from 1 to m. m is used to restrict a minimum length of the nucleotide sequence connecting the helix ensuring a valid helix. Since a base pair does not form in isolation, these rules can be useful to find all possible helices and use these to form the secondary structure. For the purposes of this research, a helix is considered only if it contains, at the very least, three adjacent canonical base pairs to form a stack, and the loop connecting the stacked pairs must be no shorter than three nucleotides in length as shown in Figure 2.1. Using these simple rules it is now easy to compute the set H of all possible helices. A valid secondary structure is the subset S of H containing all helices that make up the actual structure. Since a single nucleotide can only pair with at most one other nucleotide in a structure, the helices must not overlap. This problem is highly combinatorial as there are $2^{|H|}$ subsets of H.

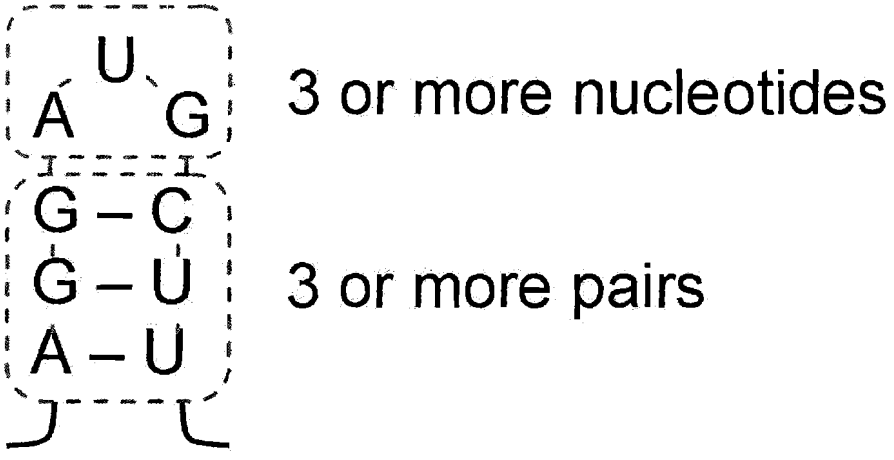

Figure 2.1: Each helix found by the helix generation algorithm must have at least three stacked pairs and the number of nucleotides connecting the stacked pair must be no shorter than three. Fig. taken from [1], page 175, permission granted by authors.

2.4.3 Tertiary structure

Tertiary structure refers to the interactions of secondary structure elements in an RNA molecule in 3D space. In this representation, each atom has a fixed coordinate in 3D space. Examples of common interactions in tertiary structure are pseudoknots, kissing hairpins, and bulge contacts. These structures form the true three-dimensional structure of an RNA molecule.

2.5 Determining structure

It is widely believed that the function of a biomolecule is largely dictated by its structure. The ultimate goal of structure prediction is to obtain the three dimensional structure of biomolecules through computation. In medicine, accurate structural knowledge would be the key to creating new lead compounds which would eventually be developed into more effective drugs. Structure-based drug design has received much attention recently. This type of research is done by increasing the understanding of molecular recognition on active sites in large biomolecules such as RNA and proteins. An experiment that is commonly done once structural information is known is flexible ligand docking [44]. Docking is a quantitative optimization technique that attempts to orient a small ligand to bind to an active site on a large biomolecule. This affects the biomolecule by either enhancing or reducing its function.

A hypothesis is that the secondary structure and tertiary structure form independently, but in sequence, of each other. The reasoning is that the thermodynamics of formation of the former are much more important than that of the latter. Separating these two enables us to treat them independently. It is also thought that in order for the tertiary structure to form, the secondary structural elements must form first as tertiary structure can be represented by interactions between secondary structure elements.

The ideas expressed in the previous section make a convenient and usable plan for structure prediction. By separating the ideas of primary, secondary, and tertiary structure, they can be solved independently. Solving the primary structure is a trivial task. Using an automated sequencer has enabled scientists to determine the primary structure with little or no user assistance. The pioneering work to make sequencing possible was done by Holley *et al.* [45].

2.5.1 Physical methods for determining structure

A simple method for solving the three dimensional structure of RNA has been elusive. Some existing methods have been applied to the domain of RNA and proteins. The two most important methods are NMR and X-ray crystallography.

NMR

Nuclear Magnetic Resonance [46, 47] (NMR) is an indirect method of structural elucidation. An experiment proceeds as follows: A sample is placed in a static external magnetic field. An antenna is used to irradiate the sample with radio waves. Different frequencies are absorbed by the sample's atomic nuclei in different chemical environments. Each nucleus absorbs radiation, and then re-emits it. A detector antenna records this energy.

Running an NMR experiment is quite involved. The required equipment is expensive. A purified sample must be available. The experiment can last several hours, often scheduled to run overnight. The resulting spectra are complex and must be analyzed by highly skilled specialists before any structural information is deduced.

X-ray crystallography

Crystallography [48] is a direct method for structure determination. The technique involves studying the pattern produced by the diffraction of X-rays through a closely spaced lattice of atoms in a crystal. The recorded diffraction patterns are analyzed to reveal the structure of the molecule. A common problem in using this method is that not all organic molecules crystallize easily, and therefore, cannot be used in this type of analysis.

A criticism of these two methods is that the environment to which the RNA molecule is subjected to for physical structure determination may not give the natural fold. For instance, in NMR, the RNA molecules being studied are exposed to a different solution as well as a different ambient temperature. In X-ray crystallography, the RNA molecules must take a particular conformation to allow for precipitation and crystallization. This conformation may not accurately represent the natural fold.

2.5.2 Energy minimization for predicting structure

RNA, like all molecules, must comply to the laws of thermodynamics. An assumption is that the natural fold is a low energy structure. Another assumption is that the contributions

of RNA secondary structure components, such as stems and loops, are independent and additive.

The search space of RNA secondary structure prediction is very large. Although enumerating and studying every possible structure for a sequence would solve the folding problem, it is not feasible. One alternative is to use creative searching techniques for energy minimization with a thermodynamic model.

RNA thermodynamics and energy minimization are discussed further in Chapters 3, 4, and 5.

2.6 Chapter summary

This chapter has introduced RNA. An RNA molecule consists of a sequence of nucleotides where each base has the ability to interact with one other base through hydrogen bonding forming pairs. RNA plays a central role in protein synthesis, but has also been found to have some catalytic properties. Its structure can be abstracted by three levels of complexity: primary, secondary, and tertiary structure. Each abstraction is more complex than the former.

RNA secondary structure can be elucidated using physical methods such as NMR and X-ray crystallography. The difficulties associated with these methods provide a rationale for attempting to predict the structure of RNA. The most common method used for structure predicting is through energy minimization.

For the purpose of the helix generation in structure prediction, this thesis defines a helix as a stack of three or more base pairs connected by three or more nucleotides. By finding which combination of base pairs, and helices, form stable structures, it may be possible to find structures similar to the natural fold.

Chapter 3

Thermodynamics of RNA secondary structure

Most prediction methods for RNA secondary structure use free energy as their metric. Simply, free energy (ΔG) is energy which is available to do useful work. Differences in free energy, in a reaction or a conformation change, provide information on process spontaneity. A negative free energy difference in a reaction favors the products and is spontaneous in that direction while a positive free energy difference favors the reactants. Free energy is often represented as a function of enthalpy ΔH (the amount of energy possessed by a thermodynamic system for transfer between itself and the environment), temperature T (a measure of the average kinetic energy in a system), and entropy ΔS (the quantitative measure of the relative disorder of a system).

$$\Delta G = \Delta H - T\Delta S \tag{3.1}$$

It is important to note that differences in free energy between two structures will also dictate the relative amounts at equilibrium.

K is the system's equilibrium constant:

$$K = \frac{[C_1]}{[C_2]} = e^{(-\frac{\Delta G}{RT})} \tag{3.2}$$

where C_1 and C_2 represent the concentration of two different structures in equilibrium in a system, R is the gas constant, and T is the absolute temperature of the system. The equation simply shows that the concentration ratio in an equilibrium varies exponentially

with free energy. The ratio follows an exponential curve where small differences in free energy have large effects on the relative concentration between two conformations.

This is important for RNA because its energy surface is not simple or smooth. The resulting surface looks very rough with many local extrema. Changing a few base pairs can greatly impact the overall secondary structure elements. Therefore, special optimization techniques using good thermodynamic parameters are needed for secondary structure prediction.

Secondary structural elements, and the sequence that form them, account for the bulk of the structure's free energy. Because RNA secondary structure is simply a list of base pairs, evaluating structures can be done by using different thermodynamic rules on these base pairs or sets of adjacent base pairs. Each structural element is independent of each other in the way the parameters are described, giving them an additive property. The summation of the individual contributions gives the total free energy of a structure [49]. The current models are not perfect since there is uncertainty in thermodynamic models. The models are incomplete and are built on noisy data. Some models are too simple and do not capture all of the free energy contributions. Other problems with current models is that they lack the parameters to correctly model some substructures. The uncertainty in the thermodynamic models translates to uncertainty in the free energy evaluation of the structures. This is why it is believed that the real structure is often a suboptimal one [50, 51, 52].

3.1 Hydrogen bond models

The simplest way to implement a hydrogen bond model is to assign a free energy change to the formation of single base pairs.

3.1.1 The Major model

According to [34], the energy value attributed to each base pairs can be made proportional to the approximate relative strength of the canonical base pairs.

ΔG(GC) = -3 kcal/mol
ΔG(AU) = -2 kcal/mol
ΔG(GU) = -1 kcal/mol
(all at 37 °C)

The basis for this choice of thermodynamic model comes from the fact that the GC pair has three hydrogen bonds, the AU pair has two hydrogen bonds, and the wobble pair GU has much weaker bonding than the AU pair [34].

According to [34], the difference in free energy can be calculated as follows:

$$E(S) = \sum_{i,j \in S} e(r_i, r_j) \tag{3.3}$$

Here, $e(r_i, r_j)$ denotes the free energy ΔG contribution between the i^{th} and j^{th} nucleotide from the formation of a base pair.

3.1.2 The Mathews model

A second model is based on the same principle of attributing energy contribution to individual base pairs. However, instead of using the approximate proportional stability of the base pairs, the number of hydrogen bonds is used.

ΔG(GC) = -3 kcal/mol
ΔG(AU) = -2 kcal/mol
ΔG(GU) = -2 kcal/mol
(all at 37 °C)

Similarly, the basis for this choice of thermodynamic model comes from the fact that the GC pair has three hydrogen bonds, the AU pair has two hydrogen bonds, and the GU wobble pair also has two hydrogen bonds.

Again, to calculate a structure's free energy, the free energy contribution (loss) from each base pair is summed [53].

3.1.3 Limitations and rationale for hydrogen bond models

These last two models were designed with the idea that thermodynamic properties of RNA secondary structure are based on the identity of each individual base pair. This is a reasonable approximation because each base pair decreases the amount of free energy in the structure. Also, these rules assume that the decrease in free energy depends only on the identity of the base pairs. This type of model is very plausible, but it fails to adequately model other intra-molecular energy contributions, such as stacking energies, loop strain, and sterics.

3.2 Stacking-energy models

Free energy of formation has been determined for many duplexes. Duplexes are two strands of RNA containing some stacked base pairs. One process [54] by which the energy can be determined is described below. First, the duplex of interest is synthesized [55] in the lab. This is usually done using standard solid-state chemistry techniques. Simply put, each strand is built on a polymer support, such as a small bead. Once the first nucleotide is attached, each subsequent nucleotide can only attach on the uncovered extremity. This way, the sequence identity can easily be controlled. When the desired sequence is synthesized, the strand is removed from the polymer-nucleotide junction. Its identity is then confirmed by NMR and its purity is confirmed by high pressure liquid chromatography (HPLC). HPLC is a standard chemistry technique used to separate mixtures. A mixture is passed in a stream of solvent (mobile phase) through some material. The components of the mixture interact with the surface of the material through adsorption. The components of the mixture have a different adsorptions and therefore different separation rate. Usually, the separation is large enough to detect the number of components in the mixture and their amounts.

Once the component is isolated and purified, the thermodynamic parameter can be determined. Chemical substances have specific frequencies where they absorb electromagnetic radiation. The amount of radiation is related to concentration of a chemical substance. Using a spectrophotometer, by changing the temperature and monitoring the absorbance, the curve of absorbance vs. temperature can be plotted. From this curve, the energy needed to break the base pairs (often termed melting) can be determined. From this energy, the free energy loss of base pair formation can be calculated.

Using data from many duplexes, it is possible to derive thermodynamic parameters for adjacent base pairs by solving sets of equations. Thermodynamic parameters have been determined in this way (and other more sophisticated ways) for almost all possible adjacent bases over the years [53]. The next sections will describe two thermodynamic models that make use of these parameters.

3.2.1 Individual Nearest-Neighbor model (INN)

In 1974, it was hypothesized that the contribution of each base pair in a helix contributes to the stability of that helix and depends on its nearest neighbors [49]. The paper describes that the enthalpy of a GC base pair will be different if it is next to an AU base pair than if

5′GG3′	5′GC3′	5′GA3′	5′GU3′	5′AG3′
3′CC5′	3′CG5′	3′CU5′	3′CA5′	3′UC5′
5′AA3′	5′AU3′	5′CG3′	5′CA3′	5′UA3′
3′UU5′	3′UA5′	3′GC5′	3′GU5′	3′AU5′

Figure 3.1: The 10 Watson-Crick nearest-neighbors.

it is next to a UA base pair. The initial study only considered Watson-Crick base pairs, *i.e.*: GC and AU. With only these two pairs, there are only 16 possible base pair adjacencies, or doublets. Due to rotational symmetry, only 10 of these are unique. These are listed in Figure 3.1. The free energy of these 10 nearest-neighbors were determined at 25°C.

The formation of a double-stranded helix can be thought of as a concentration dependent formation of the first base pair (initiation), followed by a closing of subsequent base pairs (propagation). The first pair involves hydrogen bonding only. The subsequent pairs add to this their stacking interactions. The free energy of each subsequent pair involves free energy changes which depend on sequence. Propagation is independent of concentration since it is a local intra-molecular reaction.

The author [49] warns that this data only contains parameters for Watson-Crick pairs, and does not discuss GU base pairs. He also warns that this data was generated from very similar strands. Further studies would most likely modify the parameter values.

In 1986, the 10 Watson-Crick nearest-neighbor thermodynamic parameters were re-measured at 37°C [56]. This temperature should more closely model physiological conditions. Calculating the free energy of a strand using the INN model is straightforward. Here is an example for the predicted free energy change of helix formation for $\begin{matrix}5'\text{GGCC}3'\\3'\text{CCGG}5'\end{matrix}$:

$$\Delta G^\circ_{37}(\text{pred}) = 2\Delta G^\circ_{37}\begin{matrix}5'\text{GG}3'\\3'\text{CC}5'\end{matrix} + \Delta G^\circ_{37}\begin{matrix}5'\text{GC}3'\\3'\text{CG}5'\end{matrix} + \Delta G^\circ_{37\text{init}} + \Delta G^\circ_{37\text{sym}} \tag{3.4}$$

Using the values from the table provided in the article, this becomes:

$$\Delta G^\circ_{37}(\text{pred}) = 2(-2.9) + (-3.4) + 3.4 + 0.4 = -5.4 \text{ kcal/mol} \tag{3.5}$$

The nearest-neighbor terms are generated by looking at the duplex through a window that is two base pairs wide from left to right. In this example, there is a term for

5′GG3′
3′CC5′

followed by a term for

5′GC3′
3′CG5′

followed by a term for

5′CC3′
3′GG5′

The last term,

5′CC3′
3′GG5′

is the same as

5′GG3′
3′CC5′

except for being rotated by 180°. The initiation term is a constant used to account for the loss of entropy, ΔS, during initial pairing between the two first bases. Entropy is lost because the reaction goes from two strands to a single duplex creating a more ordered system. Equation 3.1 reminds us that this is unfavorable with respect to stability; the entropy change is negative, adding a positive value to the free energy. The last term corrects for symmetry. This self complementary strand shows two-fold rotational symmetry. Again, for reasons of entropy, this destabilizes the strand. The symmetry term has a lesser effect than initiation but must be counted nonetheless.

The parameters described in [56] do not represent a complete set for use in structure prediction. Canonical base pairs also include GU pairs. Numerous examples of terminal GU pairs are found at the end of helical regions and its stacking energy was found to be approximately equal to a terminal AU pair. Other parameters determined around the same

time were those of unpaired terminal nucleotides, terminal mismatches, and parameters for internal GU pairs [56, 57].

In 1991, 11 nearest-neighbor interactions involving GU mismatches were derived from new and existing thermodynamic data [58]. The sequences included both isolated and adjacent GU mismatches. An anomaly was discovered where the thermodynamics of

5′UG3′
3′GU5′

sequences are different from those of

5′GU3′
3′UG5′

sequences. However, the most surprising result showed that the nearest-neighbor

5′GU3′
3′UG5′

in the middle of a helix in the contexts of

5′CGUG3′
3′GUGC5′ ,
5′UGUA3′
3′AUGU5′ ,

and

5′AGUU3′
3′UUGA5′

destabilize the helix. However, addition of the same

5′GU3′
3′UG5′

in the middle of

5′GGUC3′
3′CUGG5′

increases the stability.

The authors of [58] describe this as a non-nearest-neighbor effect. Whereas the nearest-neighbor

5′UG3′
3′GU5′

is always stabilizing independent of context, the

5′GU3′
3′UG5′

is dependent on context and adds corrections to previously determined parameters [56]. In this particular situation special parameters are needed to account for this anomaly [53].

It was also found that there is a very weak stabilizing effect from

5′UU3′
3′UU5′

mismatches [59]. However, since this nearest-neighbor does not contain canonical base pairs, its effects will be ignored.

A survey on stacking energies was presented in 1995 [60]. Calculations for free energy determination of different strands are presented in a straightforward fashion using nearest-neighbor parameters at 37°C. The following can be used as a guide for an implementation of the INN model.

The article [60] describes how to calculate the free energy of duplexes, and terminal mismatches. For the non-self-complementary duplex

5′CGGC3′
3′GCCG5′ ,

the calculation is described by:

$$\Delta G^\circ_{37} = \Delta G^\circ_{37} \begin{matrix} \text{CG} \\ \text{GC} \end{matrix} + \Delta G^\circ_{37} \begin{matrix} \text{GG} \\ \text{CC} \end{matrix} + \Delta G^\circ_{37} \begin{matrix} \text{GC} \\ \text{CG} \end{matrix} + \Delta G^\circ_{37\ \text{init}} \tag{3.6}$$

For a self-complementary duplex such as

$$\begin{matrix} 5'\text{GGAUCC} \\ 3'\text{CCUAGG} \end{matrix},$$

ΔG°_{37} is calculated the same way but requires the addition of $\Delta G^\circ_{37\ \text{sym}}$ for symmetry.

$$\Delta G^\circ_{37} = \Delta G^\circ_{37} \begin{matrix} \text{GG} \\ \text{CC} \end{matrix} + \Delta G^\circ_{37} \begin{matrix} \text{GA} \\ \text{CU} \end{matrix} + \Delta G^\circ_{37} \begin{matrix} \text{AU} \\ \text{UA} \end{matrix} + \Delta G^\circ_{37} \begin{matrix} \text{UC} \\ \text{AG} \end{matrix} + \Delta G^\circ_{37} \begin{matrix} \text{CC} \\ \text{GG} \end{matrix} + \Delta G^\circ_{37\ \text{init}} + \Delta G^\circ_{37\ \text{sym}} \tag{3.7}$$

These last two equations follow the same rules as equation 3.4.

For 3′ terminal unpaired nucleotides such as $\begin{matrix} 5'\ \text{GGAUCCA} \\ 3'\text{ACCUAGG} \end{matrix}$ a mismatch term is added. In this case:

$$\Delta G^\circ_{37} = \Delta G^\circ_{37}(\text{Core duplex}) + 2\Delta G^\circ_{37} \begin{matrix} \text{CA} \\ \text{G} \end{matrix} \tag{3.8}$$

For a helix containing a 5′ terminal unpaired nucleotide such as $\begin{matrix} 5'\text{AGGAUCC} \\ 3'\ \text{CCUAGGA} \end{matrix}$:

$$\Delta G^\circ_{37} = \Delta G^\circ_{37}(\text{Core duplex}) + 2\Delta G^\circ_{37} \begin{matrix} \text{AG} \\ \text{C} \end{matrix} \tag{3.9}$$

Terminal mismatches are handled by using parameters for the mismatches. The method is again the same for the example of $\begin{matrix} 5'\text{AGGAUCCA} \\ 3'\text{ACCUAGGA} \end{matrix}$:

$$\Delta G^\circ_{37} = \Delta G^\circ_{37}(\text{Core duplex}) + 2\Delta G^\circ_{37} \begin{matrix} \text{CA} \\ \text{GA} \end{matrix} \tag{3.10}$$

In 1997, a fairly large study of internal mismatches (2×2 internal loops) was performed. Stabilities were confirmed for GU mismatches [61]. However, some stability has been found in other less common mismatches, UU, CC, GA, AC, etc. Since these are not canonical base pairs and do not increase stability significantly, they have been ignored in this research.

3.2.2 Individual Nearest-Neighbor Hydrogen Bond model (INN-HB)

It was noticed that duplexes with the same nearest neighbors but different terminal ends consistently have different stabilities. The duplex with one more terminal GC pair and one less terminal AU pair is always more stable. The reason is that switching a GC pair to an AU pair in base composition decreases the number of hydrogen bonds in the duplex by one.

To account for this difference, the INN-HB model [62] also includes a term for terminal AU pairs and therefore for the base composition of the sequence.

The improved thermodynamic parameters were derived from a study of 90 duplexes of short RNA strands containing only Watson-Crick base pairs. In the INN model, the initiation parameter for duplexes with at least one GC base pair were determined, but the initiation parameter for duplexes with only AU base pairs was not determined. It has been shown that the initiation term is dependent on the identities of the two terminal base pairs. The article [62] not only provides more accurate parameters but also provides a penalty term for each terminal AU pair.

The general equation used to calculate the free energy change of duplex formation can be written in INN-HB as the following:

$$\Delta G^{\circ}(\text{duplex}) = \Delta G^{\circ}_{\text{init}} + \sum_{j} n_j \Delta G^{\circ}_{j}(\text{NN}) + m_{\text{term-AU}} \Delta G^{\circ}_{\text{term-AU}} + \Delta G^{\circ}_{\text{sym}} \qquad (3.11)$$

Each $\Delta G^{\circ}_{j}(\text{NN})$ term is the free energy contribution of the jth nearest neighbor with n_j occurrences in the sequence. The $m_{\text{term-AU}}$ and $\Delta G^{\circ}_{\text{term-AU}}$ terms are the number of terminal AU pairs and the associated free energy parameter, respectively. The $\Delta G^{\circ}_{\text{init}}$ term is the free energy of initiation.

The only change from the INN model is the addition of a $m_{\text{term-AU}} \Delta G^{\circ}_{\text{term-AU}}$ penalty when terminal AU pairs exist in a helix. Using the data tables provided by the thermodynamic model of nearest-neighbor parameters, calculating the stability of a helix is straightforward.

For a non-self-complementary duplex such as $\begin{matrix} 5'\text{ACGAGC}3' \\ 3'\text{UGCUCG}5' \end{matrix}$:

$$\Delta G^{\circ}(\text{duplex}) = \Delta G^{\circ}_{\text{init}} + \Delta G^{\circ}_{37} \begin{matrix} \text{GU} \\ \text{CA} \end{matrix} + \Delta G^{\circ}_{37} \begin{matrix} \text{CG} \\ \text{GC} \end{matrix} + \Delta G^{\circ}_{37} \begin{matrix} \text{GA} \\ \text{CU} \end{matrix} + \Delta G^{\circ}_{37} \begin{matrix} \text{CU} \\ \text{GA} \end{matrix} +$$

$$\Delta G^\circ_{37} \begin{matrix} \mathrm{GC} \\ \mathrm{CG} \end{matrix} + 1 \times \Delta G^\circ_{37} \begin{matrix} 5'\mathrm{A} \\ 3'\mathrm{U} \end{matrix} \tag{3.12}$$

For a self-complementary duplex such as $\begin{matrix} 5'\mathrm{UGGCCA3}' \\ 3'\mathrm{ACCGGU5}' \end{matrix}$:

$$\begin{aligned} \Delta G^\circ(\text{duplex}) = \Delta G^\circ_{\text{init}} + 2 \times \Delta G^\circ_{37} \begin{matrix} \mathrm{CA} \\ \mathrm{GU} \end{matrix} + 2 \times \Delta G^\circ_{37} \begin{matrix} \mathrm{GG} \\ \mathrm{CC} \end{matrix} + \Delta G^\circ_{37} \begin{matrix} \mathrm{GC} \\ \mathrm{CG} \end{matrix} + \\ 2 \times \Delta G^\circ_{37} \begin{matrix} \mathrm{A3}' \\ \mathrm{U5}' \end{matrix} + \Delta G^\circ_{\text{sym}} \end{aligned} \tag{3.13}$$

Terminal GU pairs are treated the same way as terminal AU pairs in the INN-HB model because they also have two hydrogen bonds.

3.3 Other models

It was found that no nearest-neighbor model could be exact since there was as much as a 6% difference between the free energy changes for formation of RNA duplexes with identical nearest neighbors and identical ends [55]. These results indicate that a nearest-neighbor model is good but not perfect for the prediction of helix stability. These errors cannot be removed because they are an inherent flaw of the model itself.

Some models make use of next-nearest-neighbor models. One of these models is the R-Y model. This model attributes special stability to sequences with patterns of continuous stacking. References for these models are found in [55].

3.4 Other common RNA substructures

RNA thermodynamics has been studied for more than simple helices. It is possible to create a much more complex thermodynamic model capable of modelling bulges [63, 64, 65], internal loops [66], (single mismatch [67, 68], tandem mismatches [59, 61, 69] and other internal loops [70, 71, 72]), hairpin loops [73, 74, 75, 76, 77] (tri-loops [78], tetra-loops [79, 80, 81, 82, 83, 84, 85, 86], and larger loops [87]), multi-branch Loops [88, 89], pseudoknots [90, 91, 92], and coaxial stacking [93, 94, 95, 96]. A summary of these models can be found in [8], [60], and [53]. This research investigates how accurately we can predict RNA secondary structure by using only stacking energies.

3.5 Chapter summary

A concise description of RNA thermodynamics was provided in this chapter. Free energy can be used to predict which structures are most likely to be found in the natural fold.

Two hydrogen bond models were described: Major and Mathews. These models associate free energy contributions to each base pair in a structure. The strength of the interaction is dependent on the identity of the base pair. Hydrogen bond models are simplistic and do not model stacking energies. To address this shortcoming, nearest-neighbor models, INN and INN-HB, were developed. These models associate free energy contributions to tandem pairs. The bulk of the free energy of a structure is a sum of the individual contributions.

Chapter 4

Energy minimization for RNA structure prediction

4.1 Sequences tested

Various sequences of different lengths taken from the Comparative RNA Website [97] are tested with RnaPredict. These are *Sulfolobus acidocaldarius* (1494 nt), *Homo sapiens* (954 nt), *Xenopus laevis* (945 nt), *Drosophila virilis* (784 nt), *Caenorhabditis elegans* (697 nt), *Acanthamoeba griffini* (556 nt), *Hildenbrandia rubra* (543 nt), *Aureoumbra lagunensis* (468 nt), *Haloarcula marismortui* (122 nt), *Arthrobacter globiformis* (123 nt), and *Saccharomyces cerevisiae* (118 nt). These sequences were chosen as they represent different sequence lengths and come from various genomes of organisms that are exposed to a range of physiological conditions. Because all the results follow similar trends, only the results for five sequences will be discussed in detail. These are *Xenopus laevis* (Table 4.1), *Drosophila virilis* (Table 4.2), *Hildenbrandia rubra* (Table 4.3), *Haloarcula marismortui* (Table 4.4), and *Saccharomyces cerevisiae* (Table 4.5). Data tables for additional sequences are found in Appendix A.

4.2 Correlation between free energy and correct base pairs

The starting premise in this research is that there is a strong relationship between free energy of a structure and the accuracy of the prediction. It is expected that the lower the free energy of a predicted structure, the more correct base pairs will be present. To establish the correlation between free energy and accuracy in predicting base pairs, an experiment was

Table 4.1: *Xenopus laevis* details

Filename	d.16.m.X.laevis.bpseq
Organism	*Xenopus laevis*
Accession Number	M27605
Class	16S rRNA
Length	945 nucleotides
# of BPs in known structure	251
# of non-canonical base pairs	22

Table 4.2: *Drosophila virilis* details

Filename	d.16.m.D.virilis.bpseq
Organism	*Drosophila virilis*
Accession Number	X05914
Class	16S rRNA
Length	784 nucleotides
# of BPs in known structure	233
# of non-canonical base pairs	11

Table 4.3: *Hildenbrandia rubra* details

Filename	b.I1.e.H.rubra.1.C1.SSU.1506.bpseq
Organism	*Hildenbrandia rubra*
Accession Number	L19345
Class	Group I intron, 16S rRNA
Length	543 nucleotides
# of BPs in known structure	138
# of non-canonical base pairs	1

Table 4.4: *Haloarcula marismortui* details

Filename	d.5.a.H.marismortui.bpseq
Organism	*Haloarcula marismortui*
Accession Number	AF034620
Class	5S rRNA
Length	122 nucleotides
# of BPs in known structure	38
# of non-canonical base pairs	4

Table 4.5: *Saccharomyces cerevisiae* details

Filename	d.5.e.S.cerevisiae.bpseq
Organism	*Saccharomyces cerevisiae*
Accession Number	X67579
Class	5S rRNA
Length	118 nucleotides
# of BPs in known structure	37
# of non-canonical base pairs	2

set up. Four different thermodynamic models were tested with RnaPredict (GA details in Chapter 5). For each sequence, 7010 structures were generated by running RnaPredict with the parameters shown in Table 4.6. For each of the 701 generations (0–700), the 10 lowest energy structures are examined. These parameters were chosen to maximize diversity, yet making as much progress toward low energy structures as possible (GA parameters explained in Chapter 5).

Figures 4.1, 4.2, and 4.3 shows correlation graphs for *Xenopus laevis*, *Saccharomyces cerevisiae*, and *Caenorhabditis elegans*, respectively. The graphs plot the free energy of 10 structures per generation for 701 generations for a total of 7010 structures. Figures 4.1 and 4.2 show a high correlation where a change in energy corresponds to a change in the number of correctly predicted base pairs.

The correlation coefficients for additional sequences were tabulated in Table A.1 on page 133. Figure 4.3 demonstrates imperfections within thermodynamic models. This graph shows very little correlation between free energy and the number of correctly predicted base pairs making INN-HB inadequate for predicting *Caenorhabditis elegans* structures.

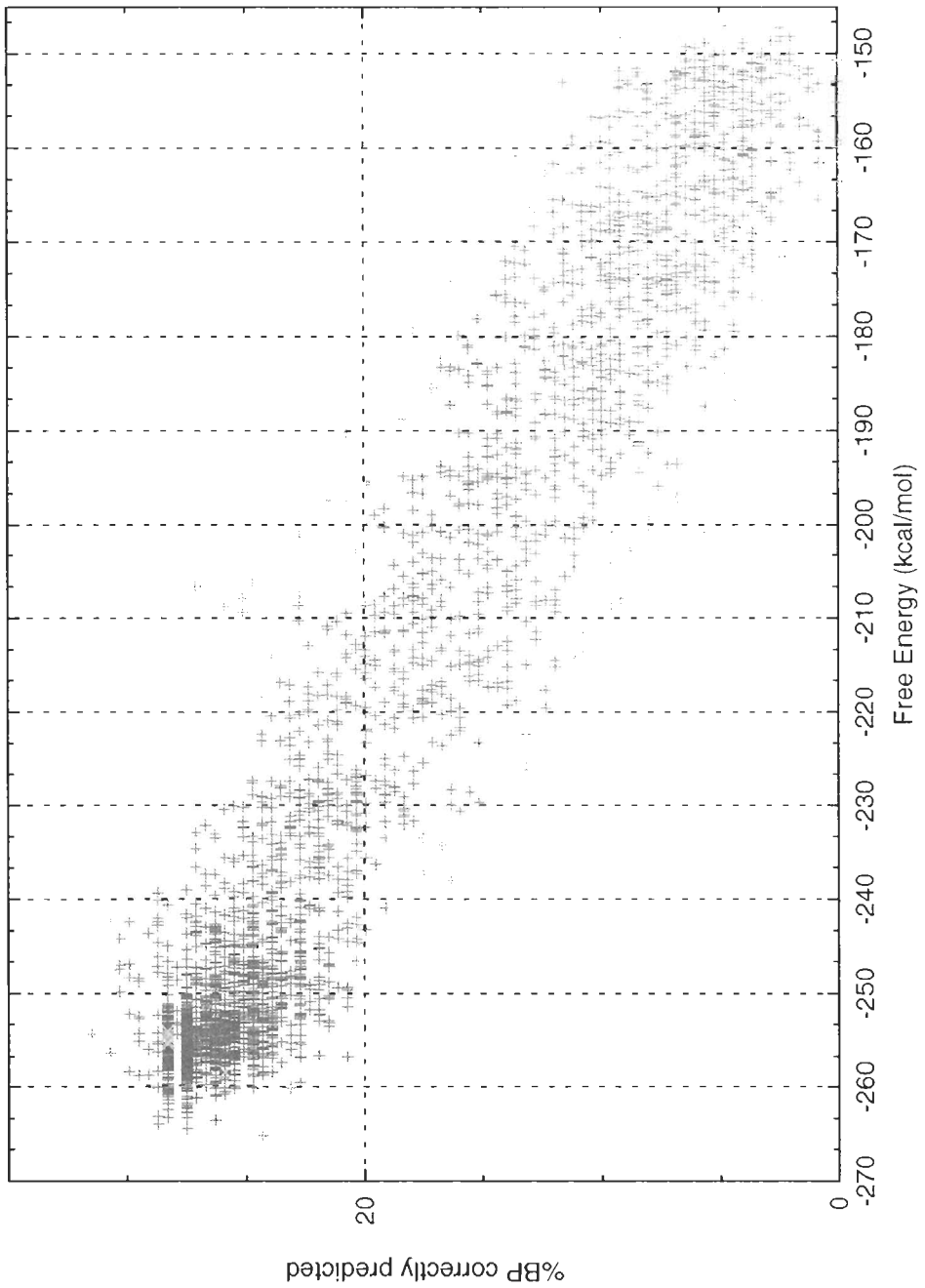

Figure 4.1: The figure shows a correlation graph for *Xenopus laevis* using INN-HB. The graph plots the free energy of 10 structures per generation for 701 generations for a total of 7010 structures. The correlation for this sequence was evaluated at $\rho = -0.96$.

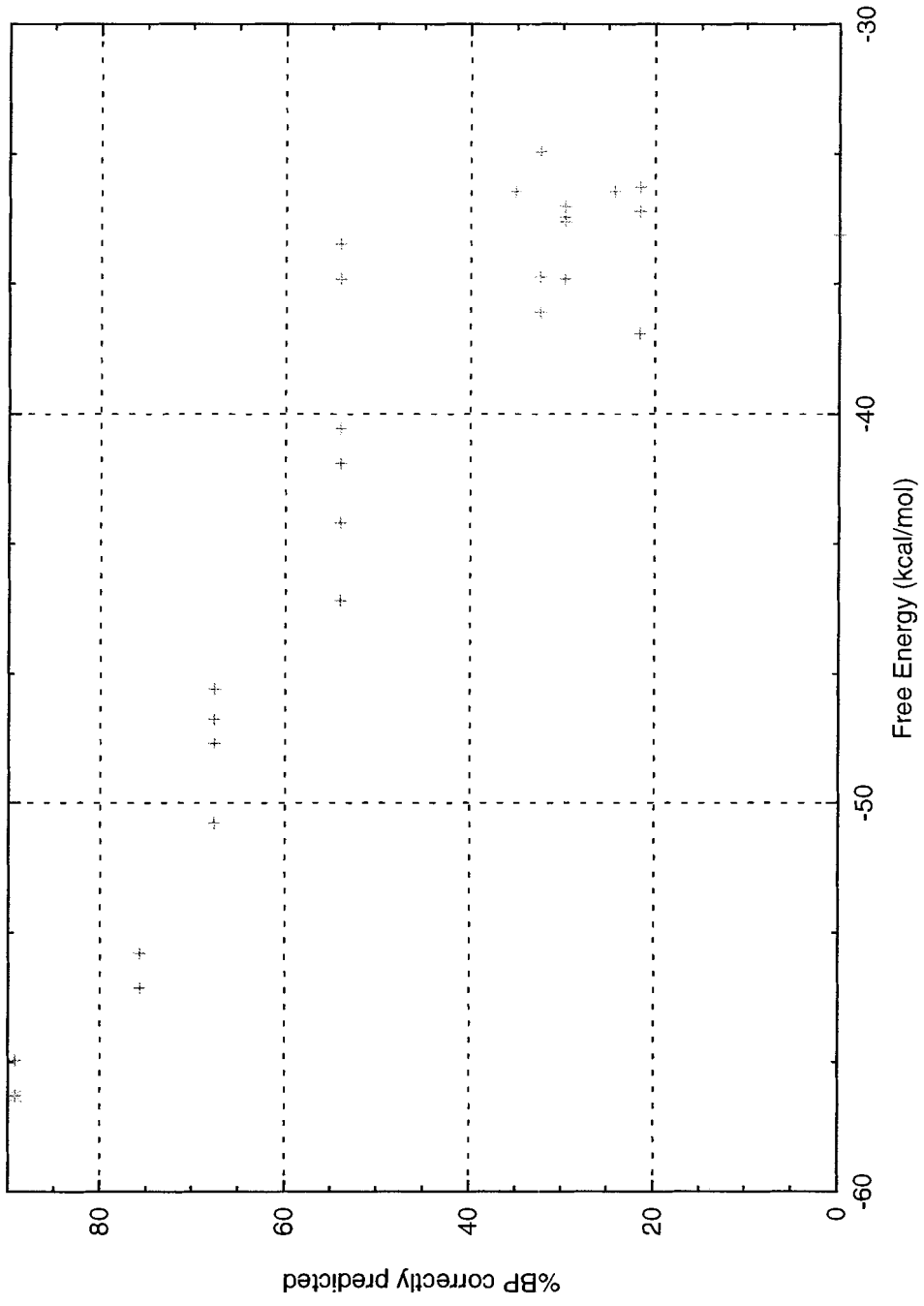

Figure 4.2: The figure shows a correlation graph for *Saccharomyces cerevisiae* using INN-HB. The graph plots the free energy of 10 structures per generation for 701 generations for a total of 7010 structures. The correlation for this sequence was evaluated at $\rho = -0.98$. Note: There are numerous duplicate structures in the population.

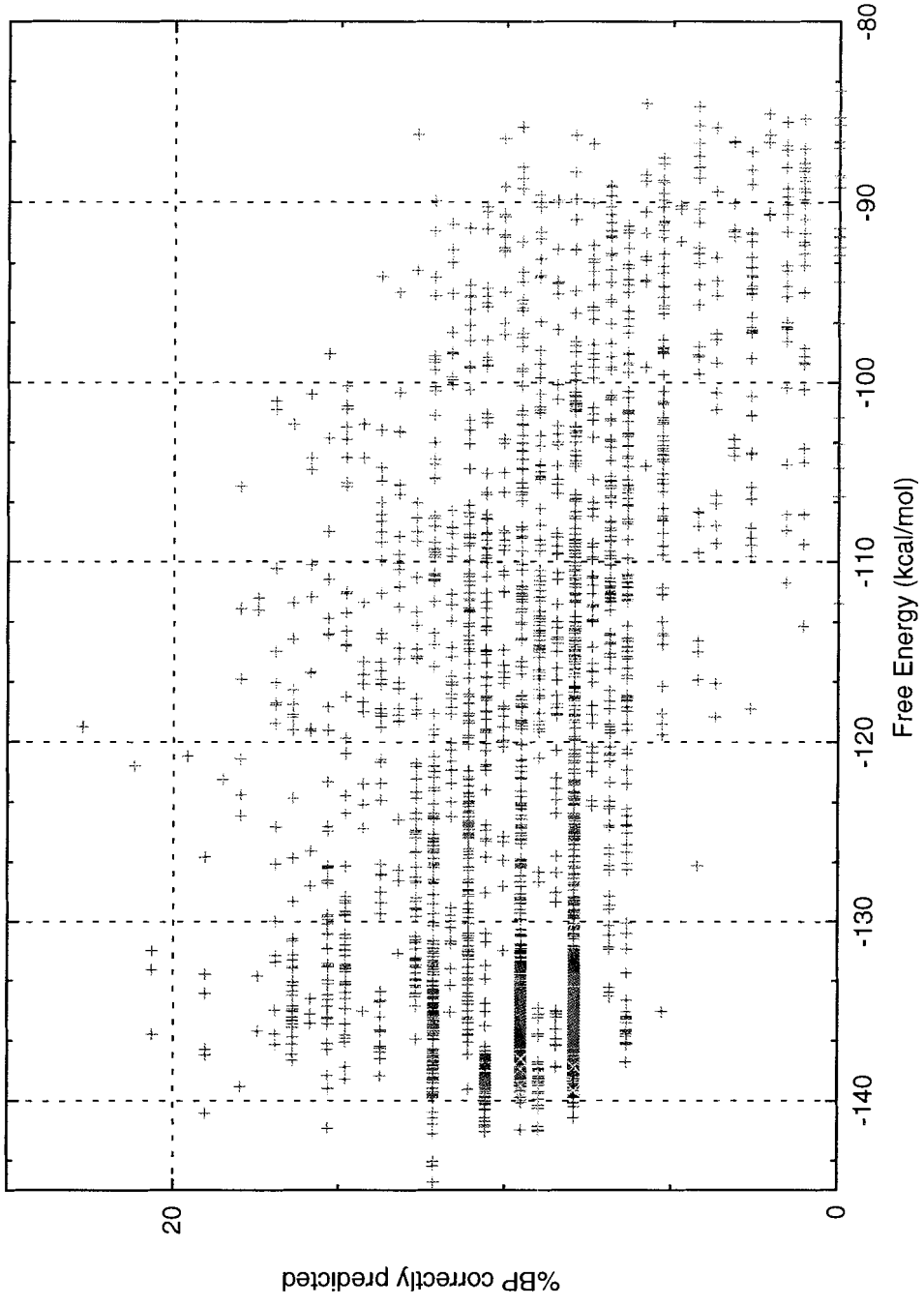

Figure 4.3: The figure shows a correlation graph for *Caenorhabditis elegans* using INN-HB. The graph plots the free energy of 10 structures per generation for 701 generations for a total of 7010 structures. The correlation for this sequence was evaluated at $\rho = -0.26$.

Table 4.6: GA parameters used to generate the correlation data.

Pop. Size	700
Generations	700
Crossover Operators	CX
P_c	0.8
P_m	0.8
Replacement	STDS
Structures per generation	10
Elitism	0
Thermodynamic Models	INN, INN-HB, Major, Mathews
Allow pseudoknots	No

Table 4.7 shows the results of the correlation between free energy of the 7010 generated structures for each parameter set and their prediction accuracy. The table shows each sequence along with the correlation coefficient for each thermodynamic model. The correlation coefficient is defined as a quantity that gives the quality of a least squares fitting to the original data [98]. For instance, in the first row, *Saccharomyces cerevisiae* shows a correlation coefficient close to -1 with INN-HB. This value shows that the lower the free energy of a structure of this sequence is, the higher number of correctly predicted base pairs are in the structure.

Table 4.7: The correlation between the free energy of structures and the number of correctly predicted base pairs.

Sequence	INNHB	INN	MAJOR	MATHEWS
S. cerevisiae	**-0.98**	-0.96	-0.15	-0.78
X. laevis	**-0.96**	-0.90	-0.78	-0.58
H. rubra	**-0.94**	-0.87	0.36	-0.71
D. virilis	**-0.93**	-0.50	-0.18	-0.71
H. marismortui	-0.74	**-0.86**	-0.56	-0.30

The data in the table shows that INN-HB yields the best correlation for all sequences except *Haloarcula marismortui.* In the latter case, INN shows a higher correlation. This data shows that stacking energy models consistently outperform hydrogen bond energy models

for these five sequences. For this reason, only the results from the stacking energy models, INN-HB and INN, will be discussed in detail in this document.

4.3 Chapter summary

Energy minimization is a valid method for prediction of secondary structures of RNA. In this chapter, the relative merit of the four different thermodynamic models was determined. This was done using a GA to generate structures to be evaluated by each model and comparing them to the known structure. Plots of free energy and number of correctly predicted base pairs were generated for each combination of sequence and thermodynamic model.

The plots give an indication on each model's ability to assess the stability of structures. Through the computation of correlation coefficients, it was shown that for most sequences, the stacking energy models, INN and INN-HB outperformed the hydrogen bond model. Furthermore, many sequences showed a strong correlation between free energy and the number of correctly predicted pairs with the stacking energy models. For the next experiments, INN and INN-HB will be used exclusively.

Chapter 5

A GA for RNA secondary structure prediction

A GA [99] is a stepwise non-deterministic algorithm that follows an evolutionary model mimicking natural evolution. It returns a number of probable solutions at each generation. In the RNA domain, a GA has the goal of finding a set of low-energy structures. At every generation in the algorithm, it is hoped that the population will contain lower energy structures than during the previous generation. By letting the algorithm run, it is expected that the population converges to low energy structures. The pseudo-code for a standard generational GA is given in Figure 5.1.

5.1 General genetic algorithm

Essentially, each generation has three key steps.

1. Random changes in the population are introduced via mutations. This step is used to avoid premature genetic convergence in the population. Randomly mutating a part of a solution tends to maintain genetic diversity within the population. Energy minimization problems can be represented by an N-dimensional hyper-surface. Using mutation helps probing different parts of the energy hyper-surface and avoids converging in local minima.

2. A combination of the parts that make up two parent solutions are chosen to make new children solutions. This is called crossover. Crossover is important to the algorithm

```
Initialize random population of chromosomes;
Evaluate the chromosomes in the population;
while stopping criteria is not reached
    for half of the members of a population
        select 2 parent chromosomes;
            apply crossover operator (P_c);
            apply mutation operator (P_m);
            evaluate the new chromosomes;
            propagation strategy;
            elitism;
        insert them into next generation;
    end for
    update stopping criteria;
end while
```

Figure 5.1: The algorithm is based on a standard generational GA. The stopping criteria is the number of generations [2].

since all solutions (member of the population) have parts that are favorable and others that are unfavorable. If a crossover is done between two of these members, it is possible that all favorable parts are incorporated into one solution and all unfavorable parts go into the other. Different types of crossover operators exist. Each one of them exhibits its own properties and heuristic.

3. The algorithm selects a new set of solutions from the old solutions. The choice is made from scoring each solution against a fitness function. This criterion selects more good solutions than bad ones which improves the overall population. All members of the population are evaluated against a fitness function and are ranked. It is the task of the GA to choose good solutions and reject others based on their scores. This way, the GA's solutions converge. Selection can act on parents, the old population, and the new population. It can be local (within a sub-population) or global (within the entire population). More details on mutation, crossover, and selection can be found in [100].

These steps can be repeated for a pre-determined number of generations, a pre-determined amount of time, or until the population converges, that is until the population's average diversity reaches a threshold value.

In summary, GAs are stochastic and non-deterministic. The initial population of solutions is generated randomly before the algorithm begins. At the mutation stage, random

parts of the solution are changed. When two parents are combined during the crossover, random parts of two solutions are exchanged. Lastly, to generate the new population for the next generation, solutions are chosen randomly where more favorable solutions are given more probability of being chosen. In this type of random algorithm, it is impossible to determine how the next step will be carried out.

In RNA secondary structure prediction, the algorithm tries to find low energy, stable structures. These are the structures that are most likely to be found naturally. Thermodynamic models associate changes in free energy to the formation of RNA substructures. In order to calculate the difference in free energy, the free energy contribution (loss) from each substructure is summed. Although it is expected that the lowest energy structure is the natural fold, it is not always so. Very often, external interactions such as solvent effects affect the resulting structure. Furthermore, the observed structure may not be the one with minimum free energy [50, 51, 52]. DPAs are at a disadvantage since they traditionally yield only one optimal structure. Since GA results yield a population of candidate solutions, it is possible to investigate not only the minimum free energy structure found but also other low energy structures that may be closer to the natural fold.

5.2 GAs for RNA secondary structure prediction

GAs were applied to the field of RNA secondary structure prediction starting in the early 1990s [28]. Since then, there have been many advances in RNA secondary structure prediction using GAs.

This type of algorithm was seen as being a very good candidate for deployment on massively parallel supercomputers. Shapiro and Navetta [28] implemented simple operators to test the viability of the GA technique. The parallel computer could compute as many as 16384 RNA secondary structures at each generation on as many processors. This first implementation performed well when compared to previous DPA techniques.

The group went on to modify the algorithm by introducing an annealing mutation operator [33]. This operator controls the probability of mutation by decreasing it linearly at each generation. This new annealing mutation operator decreases the time needed for the GA to converge and produce better results. Van Batenburg *et al.* [26, 30] proposed a modification that allowed for simulation of the folding pathway. The premise was that RNA secondary structure was influenced by kinetic processes. A method used to simulate kinetic folding

was to restrict folding to a small part of the strand where this part's size was increased after each iteration. By using this method, they simulated folding of the RNA strand during synthesis. This also helps probe local energy minima. The implementation was done in APL using bit-strings to represent the genotype [27]. Results showed structures that were more consistent with phylogenetic data than with previous minimum energy solutions.

Around the same time, Benedetti and Morosetti [29] also compared the accuracy of a GA against known RNA structures with the objective of finding optimal and suboptimal (free energy) structures that were similar. They noted that the shortcomings using the GA were not due to the algorithm itself but rather to the inadequate understanding of the thermodynamic models that influence folding.

Recently, Shapiro also modified his GA to study folding pathways [31, 101] using a massively parallel genetic algorithm.

A completely different approach was taken by Chen *et al.* [102]. Their method involves using a GA with a thermodynamic fitness function on each sequence of a related set until a certain level of stability is reached. Then, for each structure, a measure reflecting the conservation of structural features among sequences is calculated. Next, a GA is run on the structures where the fitness criterion is the measure of conservation of structural features. The resulting structures are ranked according to a measure of conservation.

Dr. Wiese's lab designed a permutation-based GA, called RnaPredict [1]. Algorithm behavior studies on the influence of selection, crossover operators, and representation issues [34, 35, 36] were also performed. A study comparing hydrogen bond and stacking energy thermodynamic models was also done [37]. The quality of the results was compared to the quality of the results from the Nussinov DPA [2]. RnaPredict was also parallelized [38, 39] improving on the serial version.

5.3 Design of RnaPredict

RnaPredict was originally designed by Dr. Wiese [1]. This GA for RNA secondary structure prediction is a standard generational GA allowing different representations, selection strategies, crossover, mutation, and elitism. The initial implementation was done in C while using some C++ features. This thesis describes the new iteration of RnaPredict which is a complete C++ reimplementation improving on all original features and adding many more. The following sections will detail how the GA was designed.

5.3.1 Representation

Traditionally, GAs are implemented using bit-strings to represent the structures in the population. Alternatively, a permutation-based encoding scheme can be used. There are several advantages to this approach. Both representations are discussed in more detail below.

Binary

RNA structures can be encoded in bit-strings. A bit-string has a length $|H|$ where H is the set of all possible helices within the helix generation model and each bit represents the presence (1) or absence (0) of a particular helix in a structure. Some helices can be mutually exclusive making a structure infeasible and thus, a repair mechanism is used to ensure only valid structures exist in the population. The repair mechanism works by reading the bit-string from left to right and changing as many bits to 0 as required to create a feasible structure. Also, because structures contain relatively few helices, the binary GA can spend a considerable amount of computational resources creating and searching infeasible structures.

Permutation

RNA structures can also be encoded with integer permutations. A permutation has a length $|H|$, similarly to binary bit-strings, but each integer corresponds to a candidate stem loop. The random population is created by generating a random permutation for each structure.

The permutation encoding proposes valid structures by reading the string from left to right, adding all stems that are compatible with stems that have already been added. By doing this, the algorithm avoids needing a repair algorithm saving on computation time and only creates and searches feasible solutions. Using this technique allows any permutation to decode to a valid structure.

Another advantage of using permutation encoding over binary encoding is that permutation crossover operators allow the GA to preserve or promote absolute position or relative order of genes. As will be shown, absolute positioning is found to have an impact on the GAs permutation-based results.

5.3.2 Selection strategies

Selection strategies are used to control breeding and survival. The operators affect which structures are chosen for recombination and also affect which are passed on to the next generation. Several selection strategies exist, such as standard, tournament, keep-best reproduction, and others. Here, we will discuss two of these: standard selection and keep-best reproduction.

Standard Selection (STDS)

STDS is described as roulette-wheel selection [103]. Each individual is given a pie-shaped slice of a wheel proportional to its fitness as compared to the sum of the fitness of all the members of the population. The roulette wheel is spun and an individual is chosen if the wheel stops on its slice. This way, highly fit individuals have a much higher chance of being chosen [34].

Keep-Best Reproduction (KBR)

The KBR operator [104, 105, 106, 107] first selects two parents via roulette wheel selection. After crossover and mutation, the best parent and best child are passed on to the next generation from a rank-based selection.

5.3.3 Binary crossover operators

Binary crossover operators are quite straightforward and easy to implement. Since they operate on single bits, or groups of bits, their execution is efficient on a computer. 1-Point, N-Point, and Uniform crossover operators are discussed here.

1-Point Crossover

The simplest binary crossover operator is the 1-Point crossover operator. This crossover operator cuts the two parents at the same, but random, position and swaps the contents of one of the segments between the parents. An example is shown below.

Suppose two parents:

```
A = 0 1 0 0 0 1 1 0 0 1
B = 1 1 0 1 0 0 1 0 1 1
```

First, a crosspoint must be chosen randomly. Suppose the crosspoint is randomly chosen at position 5.

```
A' = 0 1 0 0 0 | 1 1 0 0 1
B' = 1 1 0 1 0 | 0 1 0 1 1
```

Simply, the left hand side of the children stay the same while the right hand side is swapped between parent A' and B' (or "crossed over").

```
A' = 0 1 0 0 0 | 0 1 0 1 1
B' = 1 1 0 1 0 | 1 1 0 0 1
```

N-Point Crossover

The N-Point crossover is simply an extension of the 1-Point crossover with N randomly chosen cross-points. The parents are segmented N times and each subsequent segment is swapped between parents. Suppose the parents from the previous example are recombined.

```
A = 0 1 0 0 0 1 1 0 0 1
B = 1 1 0 1 0 0 1 0 1 1
```

Suppose two crossover points are chosen sectioning each child in three segments.

```
A' = 0 1 0 | 0 0 1 1 | 0 0 1
B' = 1 1 0 | 1 0 0 1 | 0 1 1
```

This time, every alternate segment is swapped. In this case, only the second segment is swapped.

```
A' = 0 1 0 | 1 0 0 1 | 0 0 1
B' = 1 1 0 | 0 0 1 1 | 0 1 1
```

Uniform Crossover

The third most common binary crossover operator is the Uniform crossover operator. In this case, each bit is chosen either from parent A or B by flipping a fair coin.

```
A = 0 1 0 0 0 1 1 0 0 1
B = 1 1 0 1 0 0 1 0 1 1
```

Since this is a random crossover operator, there are numerous possibilities. An example is the following:

A' = 1 1 0 0 0 0 1 0 0 1

B' = 0 1 0 1 0 1 1 0 1 1

In this example, the bits at positions 1 and 6 were changed in both parents. For the remaining positions, the coin flip returned "no-exchange."

5.3.4 Permutation crossover operators

The use of binary GAs can cause many problems in a domain like RNA secondary structure. Generating structures using a bit-string to represent the loops gives the possibility of creating infeasible structures. To avoid spending computation time on repairing infeasible structures, it is a good idea to keep track of the order of stem loop additions. A solution to this problem is adapting permutation operators from the travelling salesman problem (TSP) domain.

A lot of research has gone into the development of permutation-based crossover operators in the TSP domain. For example, Order Crossover (OX) [108], Order Crossover #2 (OX2) [109], Partially Matched Crossover (PMX) [110], Cycle Crossover (CX) [111], Edge Recombination Crossover (SYMERC) [112], and Asymmetric Edge Recombination Crossover (ASERC) [113].

Order Crossover (OX)

To discuss the OX operator, we start with two parents, A and B.

A = 8 7 1 0 6 3 4 9 5 2

B = 2 7 4 9 1 3 8 5 0 6

The substrings are swapped from one of the parents that will be designated as the donor. The substring is chosen randomly. In this example, the substring 0 6 3 of the donor A is chosen. This substring will be mapped into the receiver B.

A = 8 7 1 | 0 6 3 | 4 9 5 2

B = 2 7 4 | 9 1 3 | 8 5 0 6

First, the substring 0 6 3 is swapped out and replaced with 9 1 3. Since a total of 10 distinct elements must be maintained, the elements of the receptor that were included in the new substring must be deleted. These deleted elements are represented by ?'s for placeholders.

```
B' = 8 7 ? | 9 1 3 | 4 ? 5 2
```

To preserve the relative order of the receiver, the elements are promoted leaving the middle section intact.

```
B' = 7 ? ? | 9 1 3 | 4 5 2 8
```

At this point, the segment from the donor A is added to B to take the place of the ? giving the final permutation string.

```
B' = 7 0 6 | 9 1 3 | 4 5 2 8
```

Similarly,

```
A' = 4 9 1 | 0 6 3 | 8 5 2 7
```

The advantage of the OX is that the relative order is preserved but not so much absolute position. This property makes this type of permutation operator useful in the TSP domain since the city of origin (the first digit in the solution) is not necessarily important. In the RNA folding domain, a previously added stem loop influences subsequent additions. This type of crossover would not be a good candidate and success applying it to the RNA folding domain would be the result of chance.

Cycle Crossover (CX)

The key idea in this crossover operator is that every entry in the offspring retains a position found in one of the two parents. An example demonstrates how this operator works.

Again, start with two parent permutation strings, A and B.

```
A = 8 7 1 0 6 3 4 9 5 2
B = 2 7 4 9 1 3 8 5 0 6
```

To create an offspring, choose a parent and start. In this example, A is chosen first. The first position in the offspring is set to 8.

```
B' = 8 - - - - - - - - -
```

Since the 8 in A has been chosen as the first element, the 2 in B is no longer available to be placed in the first position of the child solution. The 2 must then be taken from it's position in A to satisfy the operator's rules.

B' = 8 - - - - - - - - 2

The 6 from B is now inaccessible so it must be added from its position in A.

B' = 8 - - - 6 - - - - 2

The 1 from B is now inaccessible so it must added from its position in A.

B' = 8 - 1 - 6 - - - - 2

The 4 from B is now inaccessible so it must added from its position in A.

B' = 8 - 1 - 6 - 4 - - 2

Since the 8 has already been added in the first step, it does not need to be added again. A cycle has been completed.

Next, copy all the remaining elements from parent B to the offspring in the same positions as the original parent B.

B' = 8 7 1 9 6 3 4 5 0 2

Similarly,

A' = 2 7 4 0 1 3 8 9 5 6

Cycle crossover works in such a way that the absolute position of each element in the child has come from one of the parents. This property makes it a viable candidate for the RNA folding domain, since the most important stem loops tend to cluster at the first few positions of a good solution. It is reasonable to assume that an operator of this kind would not produce any bad solutions from two good solutions.

Partially Matched Crossover (PMX)

A second crossover operator that was designed for order based problems such as the TSP is the PMX operator. An example is presented below:

Again, two parents, A and B, are used.

A = 8 7 1 0 6 3 4 9 5 2
B = 2 7 4 9 1 3 8 5 0 6

This time, the elements in the crossover section between the two parents are swapped. In this example, the substrings 0 6 3 of A and 9 1 3 of B are chosen.

```
A = 8 7 1 | 0 6 3 | 4 9 5 2
B = 2 7 4 | 9 1 3 | 8 5 0 6
```

First, swap 9 & 0, 1 & 6, and 3 & 3 in both parents to create new child A'. Since each element can only appear once in the final solution, any duplicates are replaced by ?'s.

```
A' = 8 7 ? | 9 1 3 | 4 ? 5 2
```

In the first permutation 9 1 was added in the middle section. Because of this, 0 6 had to be swapped out. Adding 9 1 creates duplicates in A. Originally, in A, they were in the order of 1 9 (position 3 and position 8). The original $B \rightarrow A$ was $9 \rightarrow 0$ and $1 \rightarrow 6$. In order to keep the original order and follow PMX's rules, A must replace the original position of the 1 with 6 and the original position of the 9 with 0.

This yields the following permutation.

```
A' = 8 7 6 | 9 1 3 | 4 0 5 2
```

Similarly,

```
B' = 2 7 4 | 0 6 3 | 8 5 9 1
```

The advantage of PMX is that some ordering from each parent is preserved, and no infeasible solutions are generated. It also has the advantage of preserving many absolute positions from each parent. Using PMX in the RNA domain could prove useful since there is some absolute positioning maintained.

Order Crossover #2 (OX2)

OX2 is a variation of OX. In this operator, a random number of positions are maintained from one parent while the others are copied from the other parent maintaining the same ordering. An example is used to discuss the operator. The same two parents are chosen.

```
A = 8 7 1 0 6 3 4 9 5 2
B = 2 7 4 9 1 3 8 5 0 6
```

First, a random number of indices are chosen. These indices will have their genes passed on directly to the child at the same position. In this case, indices 2, 5 and 9 are chosen from A to be passed on to the child as shown in the order vector, O.

```
O = 2 5 9
A' = - 7 - - 6 - - - 5 -
```

Next, the remaining elements are taken from B maintaining their relative order.

```
A' = 2 7 4 9 6 1 3 8 5 0
```

Similarly,

```
B' = 2 8 7 1 0 6 3 4 9 5
```

This operator has two properties. First, it keeps the absolute position of some genes from the first parent. Then, it keeps the partial ordering from the second parent. This operator should work well within the RNA prediction domain since both these properties are beneficial.

Edge Recombination Crossover (ERC, or SYMERC)

Edge Recombination Operator (SYMERC) was originally developed by Whitley *et al.* for the TSP domain. In this domain, the operator creates offspring that only contain edges from the two parent tours but may invert their direction.

An example from the TSP domain is used to explain the operator. Note that a smaller permutation will be used to illustrate how the operator works. This is simply done to reduce the size of the tables. The SYMERC operator is described as follows [112]:

1. Construct an edge table. An edge table stores all the connections from two parents that lead into and out of a city. A row in the edge table has a minimum of two entries and a maximum of four entries. Suppose the two parents:

   ```
   A = 1 2 3 4 5 6
   B = 2 6 1 3 4 5
   ```

   Example from Figure 5.2:
   2: 1 3 5 6

   The edges 2-1, 2-3, 2-5, and 2-6 exist in the parents.

1:	2	6	3	
2:	1	3	5	6
3:	2	4	1	
4:	3	5		
5:	4	6	2	
6:	5	1	2	

Figure 5.2: SYMERC edge table

2. Randomly choose an initial element from one of the two parents. This is the current city.

3. Remove all occurrences of the current city from the edge table.

4. If the current city has no more cities in its edge list, go to step 6.

5. Determine which of the cities in the edge list of the current city has the smallest edge list. This now becomes the current city. In the case of a tie, randomly choose a city to become the current city. Loop back to step 3.

6. There are no more cities in current city's edge list. Stop.

Although the above description defines the operator using terms associated to the TSP domain, the operator itself is domain independent. The operator knows only about permutations and can be applied to any domain where candidate solutions can be encoded as permutations.

Using the rules listed above, an example of crossover follows:

```
A = 1 2 3 4 5 6
B = 2 6 1 3 4 5
```

The first city is chosen randomly between from A or B. In this case the first city from B is chosen. The '2' entry is deleted from the edge table.

```
A' = 2 - - - - -
```

In row 2 of Figure 5.3, the smallest edge list from the remaining elements is found. In this case, rows, 1, 3, 5, 6 all have 2 elements and one must be chosen randomly. In this case, 5 is chosen randomly and then deleted in the edge table.

1:		6	3	
2:	1	3	5	6
3:		4	1	
4:	3	5		
5:	4	6		
6:	5	1		

Figure 5.3: SYMERC edge table

1:		6	3	
2:	1	3		6
3:		4	1	
4:	3			
5:	4	6		
6:		1		

Figure 5.4: SYMERC edge table

```
A' = 2 5 - - - -
```

In row 5 of Figure 5.4, there are two entries to chose from. Since, the edge lists from both row 4 and 6 of length 1, the tie is broken randomly. In this case, 4 is chosen and deleted from the edge table.

```
A' = 2 5 4 - - -
```

1:		6	3	
2:	1	3		6
3:			1	
4:	3			
5:		6		
6:		1		

Figure 5.5: SYMERC edge table

Row 4 in Figure 5.5 only has one element left so it is automatically chosen. This element is 3 and it is deleted in rows 1 and 4.

```
A' = 2 5 4 3 - -
```

Again, row 3 in Figure 5.6 has only one element so it is automatically chosen. This element is 1 and it is deleted in rows 2, 3, and 6.

1:		6		
2:	1			6
3:			1	
4:				
5:		6		
6:		1		

Figure 5.6: SYMERC edge table

A' = 2 5 4 3 1 -

1:		6		
2:				6
3:				
4:				
5:		6		
6:				

Figure 5.7: SYMERC edge table

Figure 5.7 has only one element left in row 1 so it is added to complete the permutation. This element, 6, is deleted in rows 1, 2, and 5 to create an empty table.

A' = 2 5 4 3 1 6

B' is done similarly.

In a modified version of SYMERC, the operator offers a mechanism to preserve common subsequences between the two parents. These are denoted as negative entries. There are three cases for an edge list:

1. There are four elements in the list. None can be negative. There are no common subsequences.

2. The list contains three elements. One element is negative representing the beginning of a common subsequence.

3. There are two elements in the list. Both are negative. This represents the internal part of a subsequence.

The modified [114] version of SYMERC gives priority to the negative entries which only affects case 2.

Asymmetric Edge Recombination Crossover (ASERC)

The Asymmetric Edge Recombination crossover operator (ASERC) was developed by Wiese *et al.* [113]. It follows the general scheme of SYMERC but not only preserves the edges but also the direction of these edges. Also, ASERC automatically preserves common subsequences.

ASERC resembles SYMERC: An edge table is constructed in a similar fashion as in SYMERC. However, only elements that have incoming edges from the current element are represented. Because of this, an edge list contains at most 2 elements. Suppose parents A and B:

```
A = 1 2 3 4 5 6
B = 2 6 1 3 4 5
```

1:	2	3
2:	3	6
3:	4	
4:	5	
5:	6	2
6:	1	

Figure 5.8: ASERC edge table

Example from Figure 5.8:

2: 3 6

The edges 2-3 and 2-6 exist in the parents. To recombine two parents, a scheme similar to SYMERC is used.

An example of crossover using ASERC follows:

The first element is chosen randomly from either A or B. In this case, 1 is chosen from A and is deleted from the edge table.

A' = 1 - - - - -

Next, row 2 and row 3 from Figure 5.9 are considered but row 3 is chosen because its edge list is shorter. The entry '3' is removed from the table.

A' = 1 3 - - - -

Only one entry, 4, remains in row 3 of Figure 5.10. It is chosen and removed from the edge table.

1:	2	3
2:	3	6
3:	4	
4:	5	
5:	6	2
6:		

Figure 5.9: ASERC edge table

1:	2	
2:		6
3:	4	
4:	5	
5:	6	2
6:		

Figure 5.10: ASERC edge table

A' = 1 3 4 - - -

Row 4 is next with a single element, 5. It is chosen and deleted from the edge table in Figure 5.11.

A' = 1 3 4 5 - -

Row 5 in Figure 5.12 contains two elements, 6, and 2. Row 6 has an empty edge and is chosen and deleted from the table.

A' = 1 3 4 5 6 -

Only one element remains in Figure 5.13 and it is added to complete the permutation. It is deleted from Figure 5.13 giving an empty edge table.

A' = 1 3 4 5 6 2

B' is done similarly.

The two ERC permutation crossover operators do not yield good results in the domain of RNA secondary structure prediction. A property of these operators is that they preserve adjacencies. Adjacencies imply that relative ordering will be maintained in the children permutations. However, in the RNA domain, relative ordering is only of minor importance and absolute positions are much more important.

1:	2	
2:		6
3:		
4:	5	
5:	6	2
6:		

Figure 5.11: ASERC edge table

1:	2	
2:		6
3:		
4:		
5:	6	2
6:		

Figure 5.12: ASERC edge table

5.3.5 Mutation

To maintain diversity in the population, a mutation operator is used. The representation affects the choice of mutation operator.

Binary

The mutation operator finds a random index in the bit-string and flips the bit. If the new bit-string is unfeasible, then the algorithm will repair it by removing all conflicting helices reading the bit-string from left to right. The new structure's free energy is then re-evaluated.

Permutation

For the permutation encoding, mutation occurs by randomly selecting two positions in a permutation and swapping their content. Next, the free energy of the corresponding new structure is re-evaluated.

5.3.6 Elitism

The elitism operator is used to pass on the fittest individuals to the next generation. The number of individuals is chosen at runtime. 1-elitism is often used to always keep the fittest individual and avoids backsliding to higher energy structures [115].

1:	2
2:	
3:	
4:	
5:	2
6:	

Figure 5.13: ASERC edge table

5.4 Computational complexity

Computational complexity is defined as the amount of resources required by an algorithm to solve a given problem. Commonly, these resources are time and space. Time complexity describes the number of steps an algorithm takes to solve an instance of the problem. This is usually defined as a function of the size of the input. Similarly, space complexity is defined as the amount of memory required by an algorithm to solve a problem. For the current discussion, only time complexity will be considered.

The time complexity of a deterministic algorithm is usually trivial to determine. For example, the time complexities of popular RNA secondary structure folding algorithms are listed:

- KinFold ($O(n^3)$) [17]
- *mfold* ($O(n^3)$) [5]
- Sankoff ($O(n^{3N})$) [19]

The previous uses **Big O** notation. This mathematical notation is used to describe the asymptotic behavior of functions. In this example, the number of steps required for KinFold to fold a sequence grows in the order of the n^3, where n is the length of the input sequence. *mfold* computation time grows similarly. The Sankoff algorithm has two inputs, the length of the sequences, n, and the number of sequences, N and thus, these two variables influence the number of steps required for computation. The Sankoff algorithm computation time increase exponentially with the number of sequences and linearly with the length of the sequences making it a very expensive algortithm.

For stochastic algorithms, the computational complexity can be more challenging to derive. In fact, GA-hardness is difficult to analyze and may not yet be formally defined [116].

A GA is essentially a set of sub-algorithms cooperating with a higher purpose. RnaPredict can be broken down into the following pieces: base pair generation, helix generation, random population generation, helix evaluation, selection, crossover, mutation, and elitism. The difficulty of complexity analysis comes from the fact that each of these algorithms has its own input which is dependent on the size of another input. For instance, it is estimated that the number of possible structures from an input of n nucleotides is larger than 1.8^n [117], and the number of possible structures in RnaPredict is $2^{|H|}$ where H is the set of all possible helices [37].

A breakdown of some of the sub-algorithms in RnaPredict and their inputs is defined:

- Sequence length - n nucleotides
- Number of possible base pairs - m
- Number of possible helices - o
- Size of a helix - p base pairs

The difficulty comes from relating these input variables. Investigating a few sub-algorithms can give some insight on computational complexity. Finding all the possible base pairs grows in $O(n^2)$ from trying to make pairs between all combinations of bases. The helix generation algorithm tries to extend each of the possible base pairs into stacks making the algorithm grow in $O(m^2)$. It is easy to see that the size of m is largely influenced by the output of the base pair generation algorithm and therefore the size of n.

Computing the free energy of a particular helix is dependent on the size of that helix and therefore grows in $O(p)$. For most of the crossover operators used, the number of steps used to perform crossover grows in $O(o)$, where o is the length of the bit-string/permutation. Decoding is done using a greedy operator reading the bit-string/permutation from left to right and therefore also has a time complexity of $O(o)$.

Calculating the time complexity of RnaPredict requires finding relationships between all the variables listed above making this a difficult task. It may be sufficient to say however that the search space has an exponential relationship with sequence length.

5.5 Implementation of RnaPredict

The original implementation of RnaPredict was done by Dr. Wiese with the help of some research assistants. There were two code bases. The first code-base was a permutation GA implementation in C using some C++ constructs totalling 2658 lines including comments. The binary GA was done similarly with a total of 1996 lines of code.

The re-implementation of RnaPredict was done by a small group of researchers. It was deemed necessary to rethink the design of the GA to make it more intuitive and maintainable. The code was written in a cleaner fashion adhering to a universal coding standard. It was also important to redesign RnaPredict to allow easy addition of new features. The implementation was done in object oriented C++ with heavy use of system calls to the standard Unix libraries and the C++ Standard Template Library (STL). The target platform for the code is Linux using the GNU C++ compiler (G++), but care was taken to code with portability in mind. The code-base used to generate the results for this thesis totals 18287 lines of code including comments. In this code, only the original code for the ERC crossover operators was included and then optimized for efficiency and adherence to the coding standard. The rest was written from scratch using the literature as a guide.

Andrew Hendriks and Edward Glen contributed largely to the design and the implementation of key modules. My contributions were 9472 lines of code, Andrew Hendriks contributed 5039 lines of code and Edward Glen added 3788. My main contributions included the command line parser, the permutation encoding and crossover operators, the front-end used to launch the GA, the GA constants, the output formatter used to write results to file, the RNA constants, and the thermodynamic models. Also, we have contributed to each other's modules.

Figure 5.14 shows a high-level Unified Modeling Language (UML) diagram of our initial redesign of RnaPredict. The diagram shows how RnaPredict was designed with low coupling between classes and high cohesion. Furthermore, RnaPredict was designed to be composed of four major abstractions: *Controller*, *Domain*, *Representation* and *GA Mechanics*. This allows for greater flexibility when writing code. For instance, most code related to RNA is found in the *Domain* abstraction while code related to the encoding (crossover and mutation) is found in the *GA Mechanics* abstraction. System calls for reading and writing data are found in *Controller*. Lastly, the GA's population is controlled by *GA Mechanics*.

If there is ever a need to adapt the current code-base to another domain, most of the

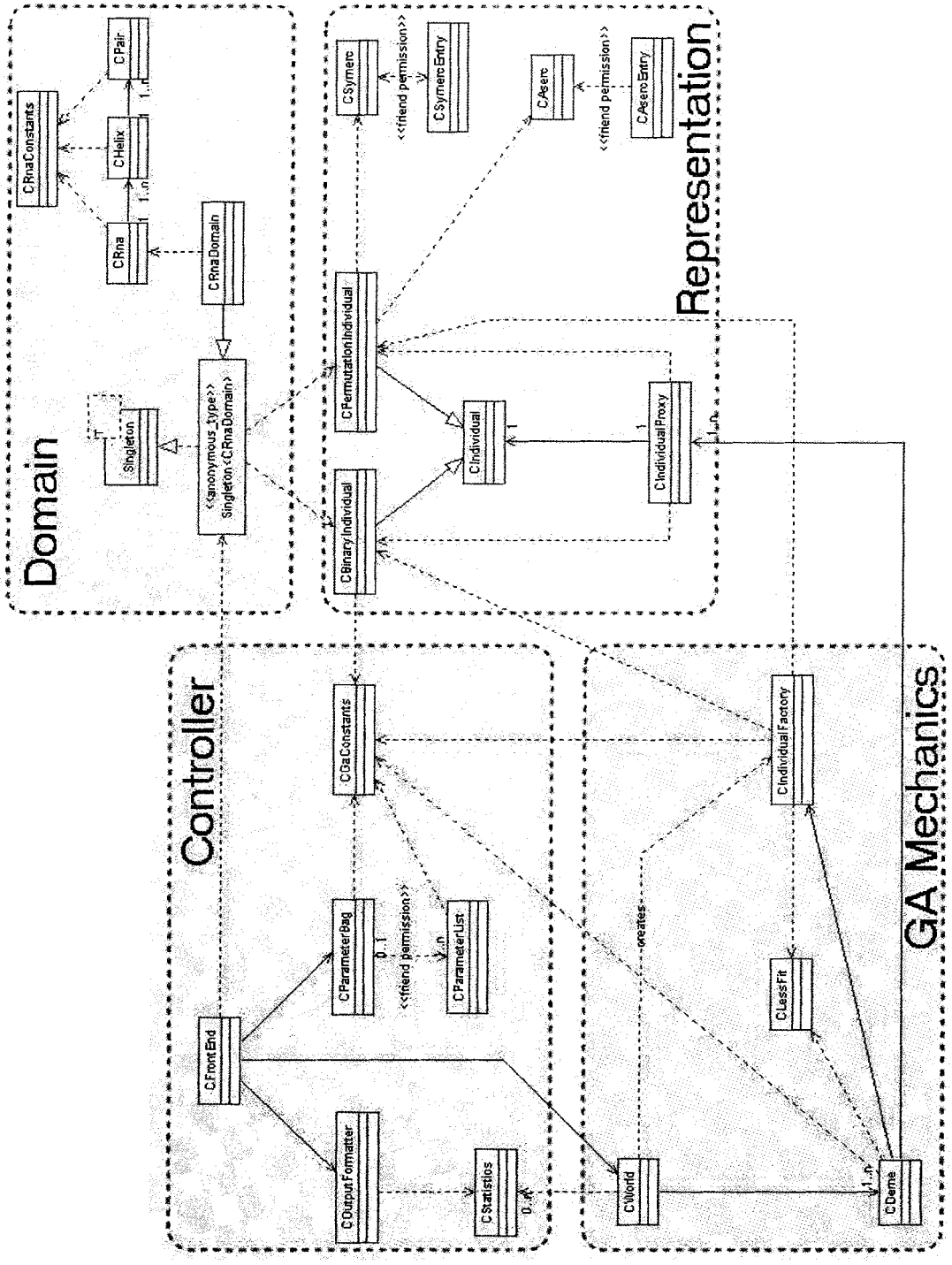

Figure 5.14: This figure shows the UML design used to guide implementation of RnaPredict. The highlights are low coupling between classes and high cohesion within classes.

code could be reused and only the Domain abstraction would need to be rewritten.

5.5.1 Source code management

Source code for the current incarnation of RnaPredict was written by Andrew Hendriks, Edward Glen, and Alain Deschênes, but was logically based on a previous implementation in C and C++. Concurrent Version System (CVS) was used to manage source code development. This system allowed the developers to collaborate more effectively and keep each developer's code synchronized with the latest source tree. CVS also allowed the developers to work on the same classes and even the same files. Using the very detailed code specifications, as long as two developers did not modify the same line before synchronizing with the main tree, conflicts were avoided.

5.6 Testing RnaPredict

Because of the use of an object oriented paradigm, it is possible to test each class before integration into RnaPredict as a whole. Each class has its own test driver where class instances can be tested thoroughly. Furthermore, the use of assert statements was encouraged. Assert statements are used to help catch logic errors quickly and effectively. Simply put, these are boolean statements where conditions can be tested at any point in the code. For instance, these statements can be useful to verify the integrity of the data before or after a function call. Asserts can require complex computations and tend to slow the execution of the code. For production code, a compiler switch to disable these statements was used for maximum performance.

RnaPredict repeats large sections of code for each generation. Unoptimized code can have adverse effects on runtime performance but these are sometimes challenging to find. Code profiling detected these bottlenecks and guided the optimization efforts.

5.6.1 Cluster computing

With the number of parameters that can be changed in RnaPredict, testing can become computationally expensive. To make matters worse, a single run typically takes a few hours but can easily reach many days for long sequences. For a sequence of approximately 1000 nt, a single run can take as much as ten hours using ASERC, but takes only fifteen minutes for a

1-Point run with 700 generations. In order to obtain a large set of results in a timely manner, it is essential to make simultaneous use of as many computers as possible. Distributing the processes manually is a time-consuming and inefficient solution.

Recently, there have been advances in distributed computing and clustering using commodity hardware. One of these clustering technologies is MOSIX [118]. From the MOSIX site:

> "MOSIX is a software package that can make a cluster of x86 based Linux servers and workstations (nodes) run almost like an SMP. The main advantages are simplicity of use and near optimal performance, e.g., when you create processes, MOSIX will assign (and if necessary reassign) your processes automatically and transparently to the best possible nodes, to maximize the performance.
>
> The core of MOSIX are adaptive management algorithms that monitor and respond to the resource requirements of all the processes vs. the available, cluster-wide resources.
>
> The algorithms of MOSIX are decentralized, each node is both a master for processes that were created locally and a server for processes that migrated from other nodes. The MOSIX algorithms are geared for maximal performance, overhead-free scalability and ease-of-use."

An open source effort to develop an alternative to the proprietary MOSIX package, called OpenMosix has been developed. Most of the results generated during the course of this research were generated using OpenMosix after switching from MOSIX early on. OpenMosix quickly showed high maturity, excellent stability, and a large developer and user base.

For this research, the data was generated using a 128 node OpenMosix Linux Cluster. This cluster is composed of standard (patched with OpenMosix kernel patches) Redhat Linux 8.0 nodes on a Gigabit network.

5.7 Data storage and analysis

5.7.1 Scripting

To make the experiments more manageable, a series of scripts were developed to automate process assignment to nodes, to verify the progress of each run, to get computer information on a node, to generate parameters automatically, to sort the nodes by performance, and to sort the runs by the resulting minimum free energy structure. A description of the various scripts follows:

bpseq2ct.sh: The known structures available at the Comparative RNA Website are in a format called BPSeq. This format resembles the CT format but lacks some information. This script converts from the BPSeq format to the CT format.

checkprogress.sh: While RnaPredict is running, there is no easy way to check how much progress is made and how long the runs still have before they finish. This script, mostly authored by Edward Glen, parses the result file estimating the amount of time left along with a percentage bar graph.

clean_results.sh: The results from an experiment are stored in a directory tree in the file system. The tree is structured as follows. First, a directory entry is created for each sequence. Within each sequence directory, a sub-directory is created for each parameter set. Within a parameter set directory, the lowest energy structure file at the end of the run along with the averaged GA statistics and graph, and comparison data is found along with directories for each random seed. Within each random seed directory, the structure file at the end of the run is found along with the GA statistics, the graph, and the results of the comparison to the known structure. When testing data generation and analysis scripts, it is sometimes necessary to clean this directory tree. Cleaning the tree removes all files except those generated by RnaPredict.

collate_results.sh: This script was primarily written by Edward Glen. It calculates the mean and standard deviation of the energy of the structures in the population for all random seeds for a particular run. It also determines which single structure has the lowest energy at the end of the runs.

compare_ct.sh: This script compares a predicted structure to the known structure. It reports the number of correctly predicted base pairs.

convertspace2underscore.sh: Working with directories containing spaces can be difficult in the Unix-like operating system. This script substitutes all spaces, in each directory, with underscores.

ct2sequence.sh: This script takes a BPSeq or CT file and extracts the RNA sequence.

fix_graph_layers.sh: When graphing with GnuPlot, the grid-lines appear behind the graph. This script brings the grid-lines to the front where they should be.

generategraphs.sh: This script goes through each run in an experiment and graphs the GA's statistics in GnuPlot. A call to the script to bring the grid-lines to the front was added.

graphstd.sh: This script graphs the GA statistics in GnuPlot. It correctly determines the number of generations and the range of energy values required as well as the optimal ranges for axes. A small internal script was added so graphs would be outputted in shades of gray, ready for publication.

mosixrun_nops.sh: This script is used to manage the runs on the cluster. It tries to send one process per node and attempts to send new runs as older runs terminate. This has the effect of keeping the cluster running at full load for as long as there are still runs in the queue.

dna2rna.sh: Some sequences are only available as DNA sequences. These sequences must be translated to RNA. This script translates a DNA sequence, to RNA, strips all white-space characters, and converts the sequence to uppercase characters.

restart_run.sh: The computer cluster and OpenMosix software have rare, but random unexplainable instability issues. When this happens, some runs are terminated prematurely. This script can be used to re-start the run without having to remember the command used

to invoke it.

5.7.2 Java programs

jRnaCompare.jar: jRnaCompare is a command-line front-end for the jViz.Rna package that allows to compare RNA structures quickly. It takes three arguments: the known structure ct file, a file containing a list of predicted structure files, and a file containing a list of output files where the statistics are written for each comparison.

5.8 Chapter summary

This chapter described how a generational GA using mutation and crossover could be used to predict the secondary structure of RNA. A literature review described how GAs have evolved in this domain.

RnaPredict is introduced as a GA able to use both binary and permutation encoding to predict the secondary structure of RNA. Global (STDS) and local (KBR) selection strategies are described to show how they could be beneficial in controlling the convergence velocity of the GA. Permutation crossover operators (CX, OX, OX2, PMX, ASERC, and SYMERC) and binary crossover operators (1-Point, N-Point, and Uniform) are detailed showing their strengths and weaknesses in this domain. In this domain, permutation encoding using CX and OX2 is expected to be beneficial due to the property of maintaining absolute positions. Mutation operators were designed to be used with their respective encodings. Elitism is used to ensure the best individual survives after each generation.

Programming practices used by the three developers, Alain Deschênes, Edward Glen and Andrew Hendriks, are described. The C++ source code was managed using CVS. Assert statements as well as unit tests were used to minimize the number of errors in the code.

Due to the stochastic nature of GAs, RnaPredict was run on a Linux load balancing OpenMosix cluster. This cluster allowed 128 single runs to be done simultaneously mapping one process to each of the 128 nodes.

A large set of BASH scripts and a java program was written to facilitate cluster management, file conversion, and data analysis.

Chapter 6

Optimization of GA parameters

An extensive set of runs was done to determine optimal settings in the GA (Table 6.1). In all, nine crossover operators were tested. CX, OX2, PMX, OX, ASERC, and SYMERC, were used with permutation encoding and 1-Point, 2-Point, and Uniform, were used with binary encoding. Both binary encoding with mutation alone and permutation encoding with mutation alone were also run as control experiments. Two selection strategies, STDS and KBR, were tested. For comparison of the relative merits of parameter choices, only INN-HB was used for the thermodynamic model. It is assumed that the thermodynamic model would not affect the GA's general convergence behavior by a large margin. Each experiment was tested with 30 random seeds to ensure statistically significant results.

6.1 GA parameters

Table 6.1 lists the parameters tested. The graphs in this chapter will track the lowest free energy structure after averaging 30 randomly seeded runs. The results in these experiments were found to be similar across different sequences.

6.2 Convergence behaviour

Figure 6.1 shows a typical experiment for *Hildenbrandia rubra*. The lighter outer envelope of the plot represents the extremities of each generation (members with maximum and minimum energies). The darker inner envelope shows the mean free energy of the population with standard deviation. This particular graph is an experiment of 30 averaged runs for

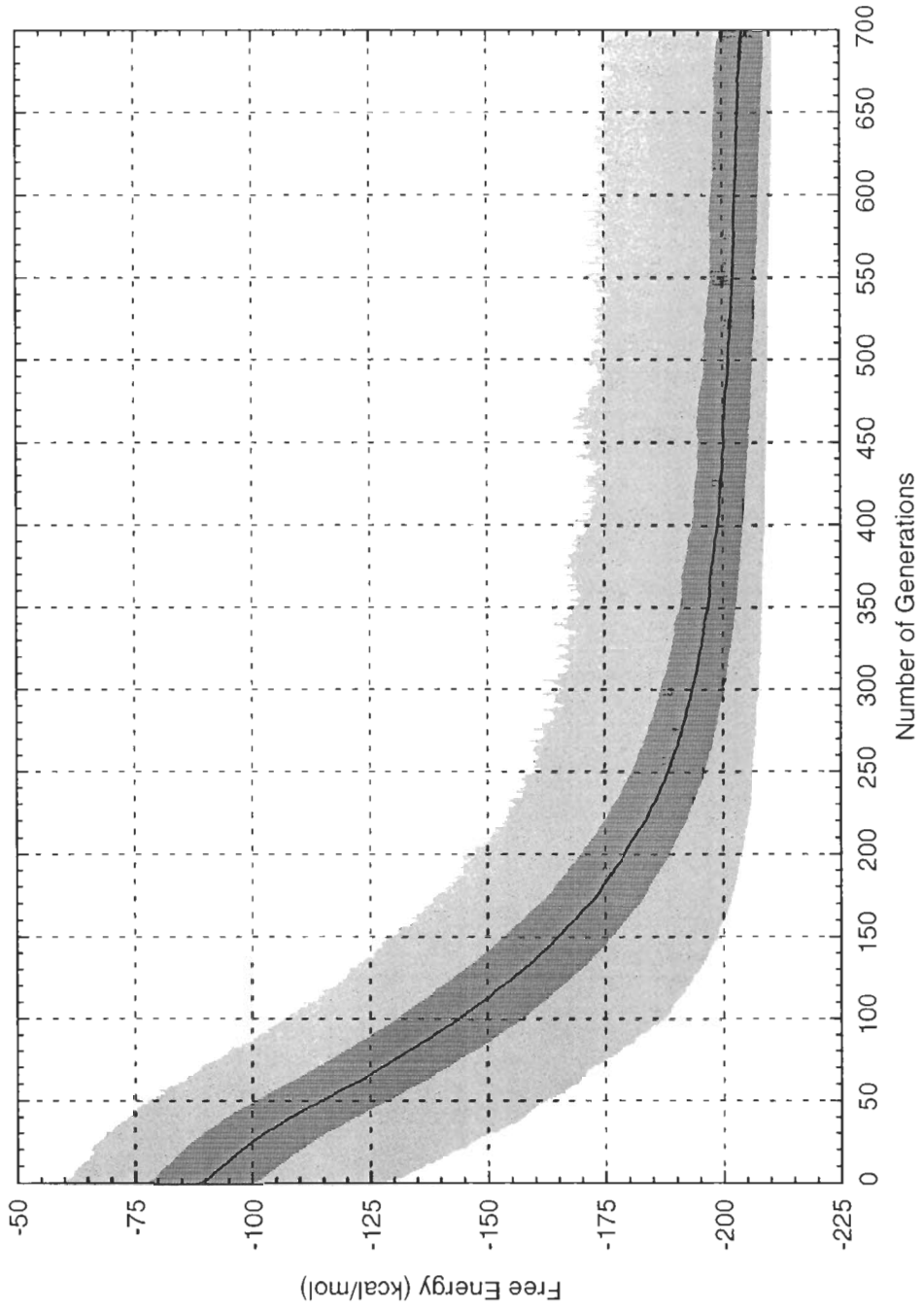

Figure 6.1: *Hildenbrandia rubra*, $P_m = 0.8$, $P_c = 0.7$, population size = 700, CX, STDS, 1-elitism, average of 30 random seeds using INN-HB as the thermodynamic model.

Table 6.1: Genetic algorithm parameters

Pop. Size	700
Generations	700
Crossover Operators	CX, OX2, PMX, OX, ASERC, SYMERC, Permutation with no crossover, 1-Point, 2-Point, Uniform, and Binary with no crossover
P_c	0.7
P_m	0.8
Selection	STDS, KBR
Elitism	1
Thermodynamic Models	INN-HB
Random seeds	30
Allow pseudoknots	No

700 generations. The graph, for the first 225 generations, shows a rapid decrease in the average free energy of the population. After this point, the graph's slope increases slowly but there is still a strong tendency for lower energy structures at each generation even after 700 generations while there is still high diversity in the population shown by the large standard deviation.

6.3 Relative merit of crossover operators

Crossover operators are used to recombine two parents to obtain two children. Three binary crossover operators are used with binary encoding: 1-Point, 2-Point and Uniform crossover operators. There are also six permutation crossover operators to be used with permutation encoding: CX, PMX, OX, OX2, SYMERC, and ASERC. The following discusses the relative merits of encoding and crossover operators.

6.3.1 Binary

1-Point, 2-Point, and Uniform crossover were used for binary encoding. A fourth binary encoding result was added without crossover but with mutation alone.

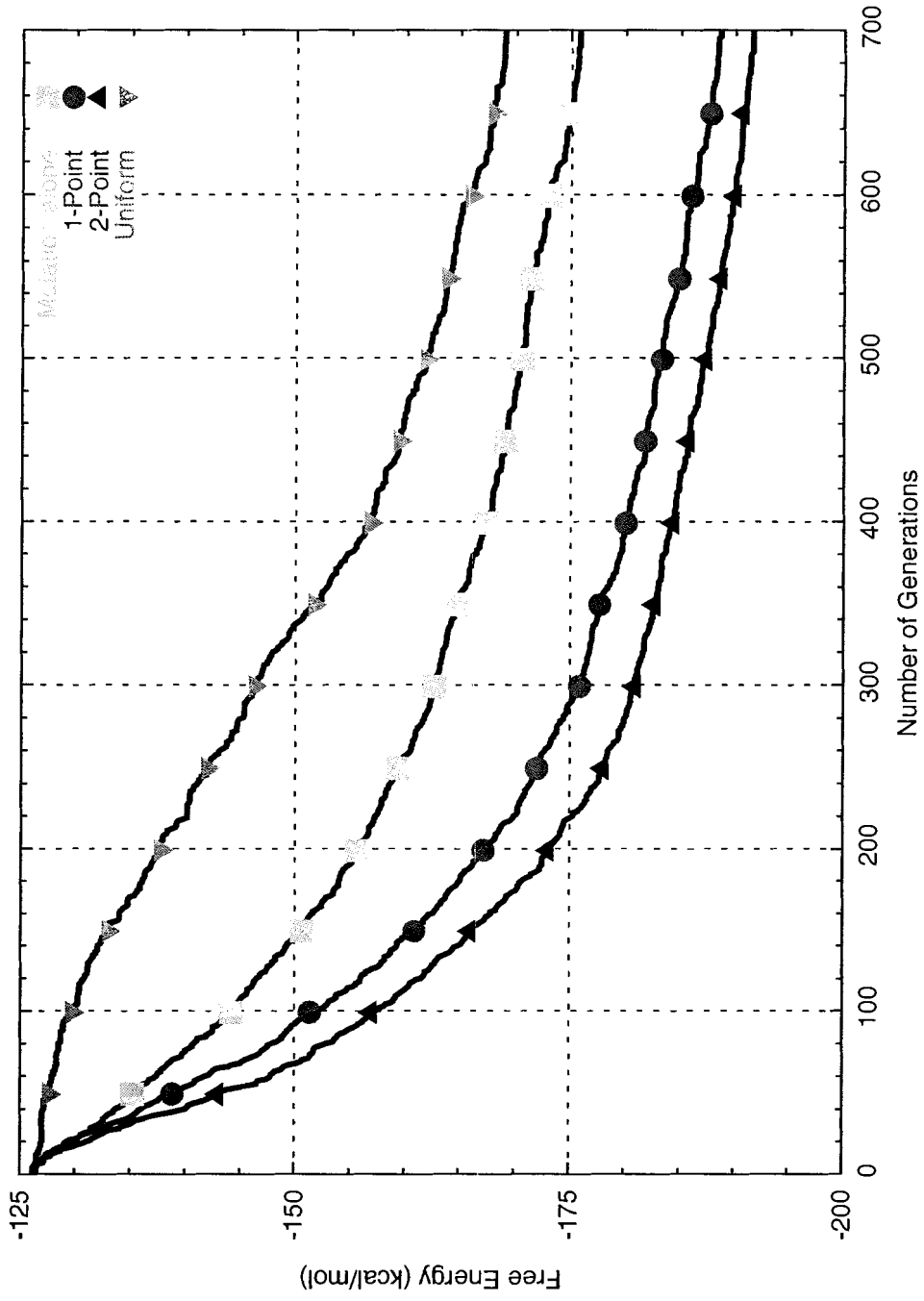

Figure 6.2: This graph compares the behavior of the different binary crossover operators with *Hildenbrandia rubra* using STDS. The graph follows the lowest energy structure from an average of 30 random seeds.

Figure 6.2 compares the behavior of the binary crossover operators. Starting with the same random population, the graph shows the average lowest free energy structure for 700 generations. Markers are placed at every 50 generations to make the interpretation easier.

The graph shows that the best crossover operator is the 2-point crossover operator. In the first 50 generations, the slope of the graph is steep and RnaPredict finds lower energy structures at each generation of the GA. After 50 generations, the slope increases slightly but is still quite steep. At generation 300, the slope increases slightly again but the GA is still quite aggressive in finding lower energy structures at each generation. RnaPredict continues to find lower energy structures even at the last generation. The 1-Point crossover operator ranks in second place and also performs well finding low energy structures. The binary experiment with no crossover was placed in the graph as a control showing the benefits of mutation alone. The worst crossover operator, Uniform, shows slow progress even after 700 generations.

Figure 6.3 compares the behavior of the same crossover operators with KBR instead of STDS. Starting with the same population, the graph shows that 1-Point and 2-Point perform equally. These two curves are very aggressive in finding low energy structures until generation 100. After this point, the graph levels off making no further progress. The same is seen with Uniform and mutation alone where they level off at a higher free energy after 100 and 200 generations, respectively.

The results from the binary crossover operators show that both 1-Point and 2-Point crossover operators outperform Uniform crossover and mutation alone. This can be explained from the fact that the cut-and-paste 1-Point and 2-Point crossover operators are able to transmit building blocks to the next generation. It is possible, however, that Uniform crossover functions more like random mutation where most of the progress toward lower energy structures is done by selection. These results are consistent across different sequences.

6.3.2 Permutation

The relative merits of permutation crossover operators were also studied. The results are shown in Figure 6.4. The results show that both OX2 and CX are clearly superior under these parameter settings. Both are very aggressive in finding lower energy structures in the first 200 generations. Beyond this point, both crossover operators continue to make progress with OX2 finding the lowest energy structures after 700 generations.

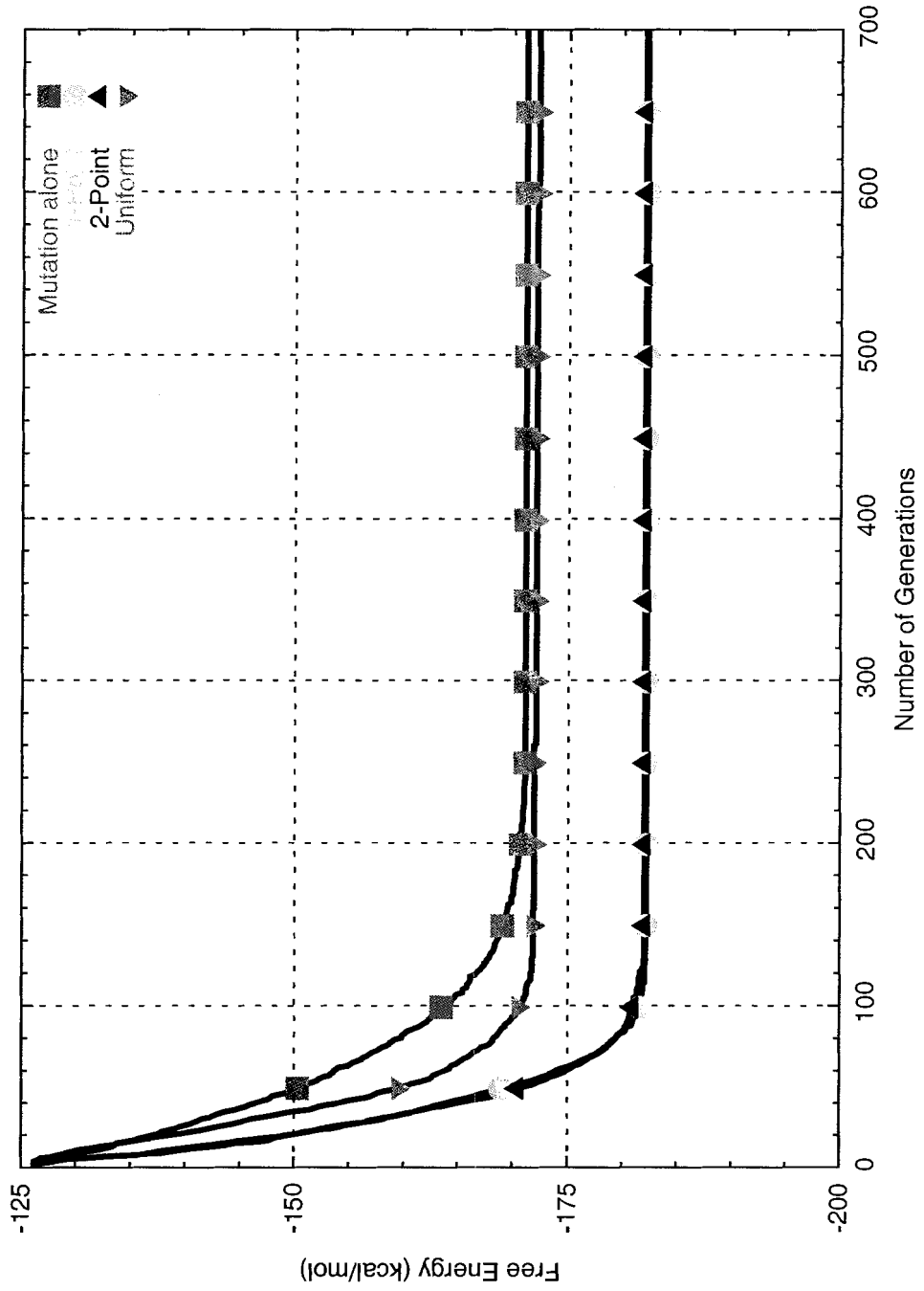

Figure 6.3: This graph compares the behavior of the different binary crossover operators with *Hildenbrandia rubra* using KBR. The graph follows the lowest energy structure from an average of 30 random seeds.

PMX and mutation alone make steady progress and show similar results throughout the 700 generations. Meanwhile, OX, ASERC, and SYMERC make very little progress with these settings and yield inferior results than mutation alone.

With KBR (Figure 6.5), the crossover operators are more competitive. This time, CX manages to edge out OX2 by a small margin after 700 generations. Although OX2 takes an early lead toward low energy structures, PMX manages to yield similar results by making steady progress throughout the 700 generations. Next, mutation alone and OX show similar results with mutation alone getting a slight edge. The two ERC operators perform equally well but relatively poorly with SYMERC taking a slight lead over ASERC in the last few generations.

CX and OX2 are the overall best permutation crossover operators in this domain. This can be partially explained due to the fact that these operators transmit absolute position of genes in recombination. In contrast, operators that would be more successful in the TSP domain, such as the ERC operators, transmit adjacencies, relative ordering, and subsequences since tours can be shifted to accommodate any starting city. These results are consistent across different sequences.

6.3.3 Binary vs. permutation

Figure 6.6 shows the relative merits of the encoding by comparing the best three permutation crossover operators, OX2, CX, and PMX with the three binary crossover operators, 1-Point, 2-Point, and Uniform using STDS.

The graphs clearly show that under these settings permutation encoding performs better than binary encoding with OX2 and CX giving the lowest energy structures. Next, 1-Point and 2-Point yield results that are better than PMX under these conditions. However, Figure 6.5 did show that PMX gave excellent results with KBR equalling those of OX2 under those conditions.

In this domain, it can be concluded that permutation encoding yields superior results than binary encoding and the best crossover operators are OX2, CX, and PMX. These results were found to be consistent across different sequences.

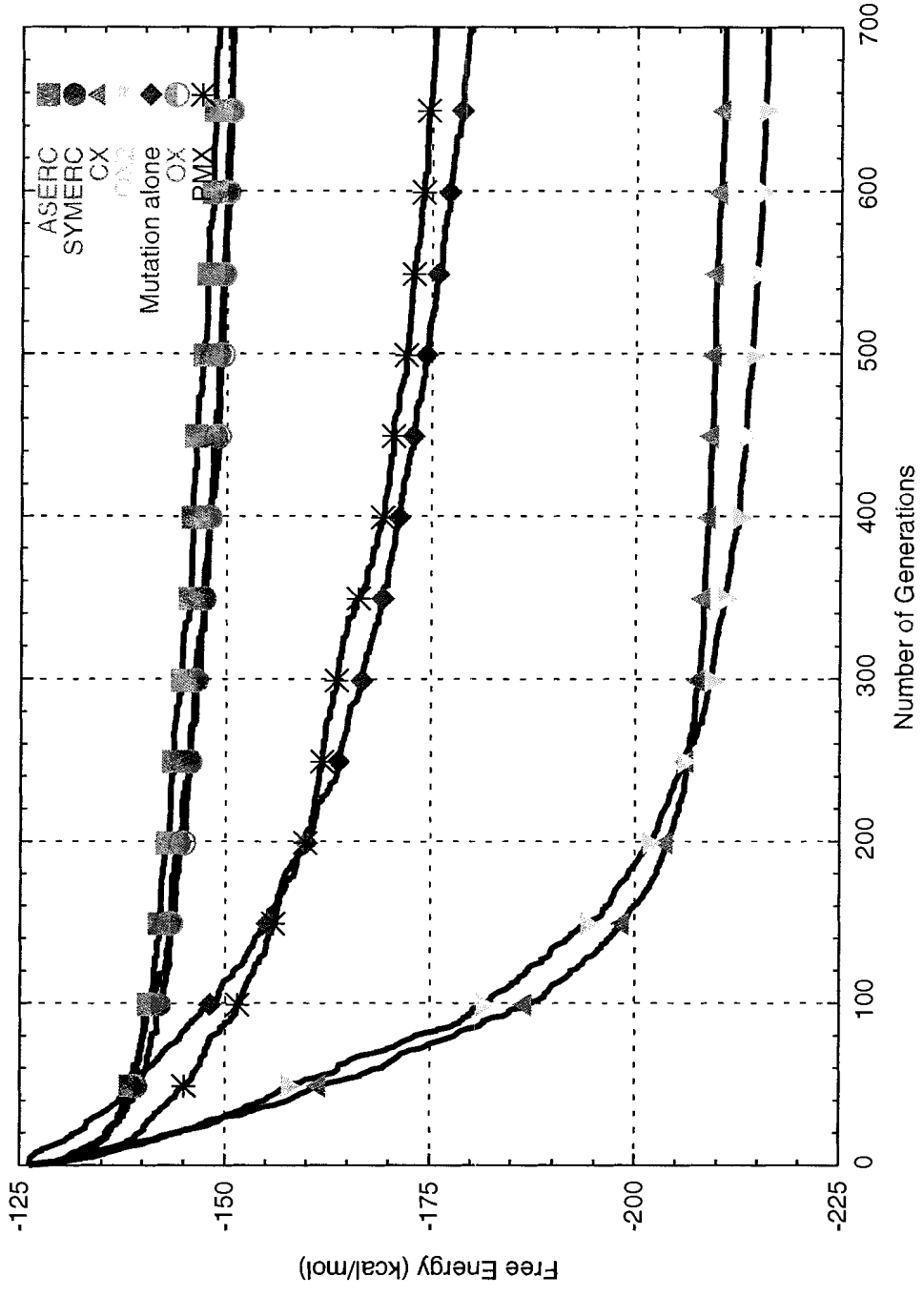

Figure 6.4: This graph compares the behavior of the different permutation crossover operators with *Hildenbrandia rubra* using STDS. The graph follows the lowest energy structure from an average of 30 random seeds.

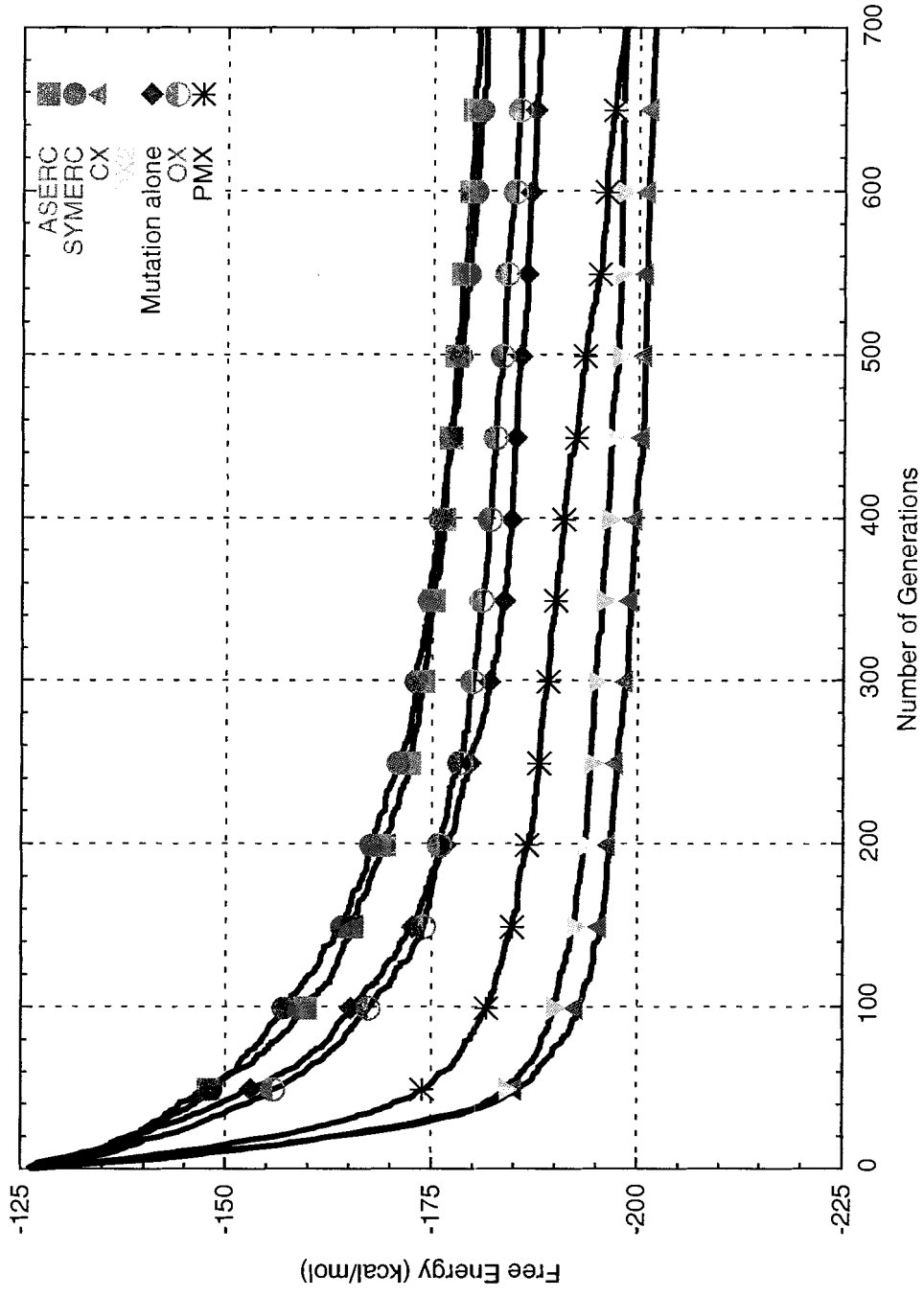

Figure 6.5: This graph compares the behavior of the different permutation crossover operators with *Hildenbrandia rubra* using KBR. The graph follows the lowest energy structure from an average of 30 random seeds.

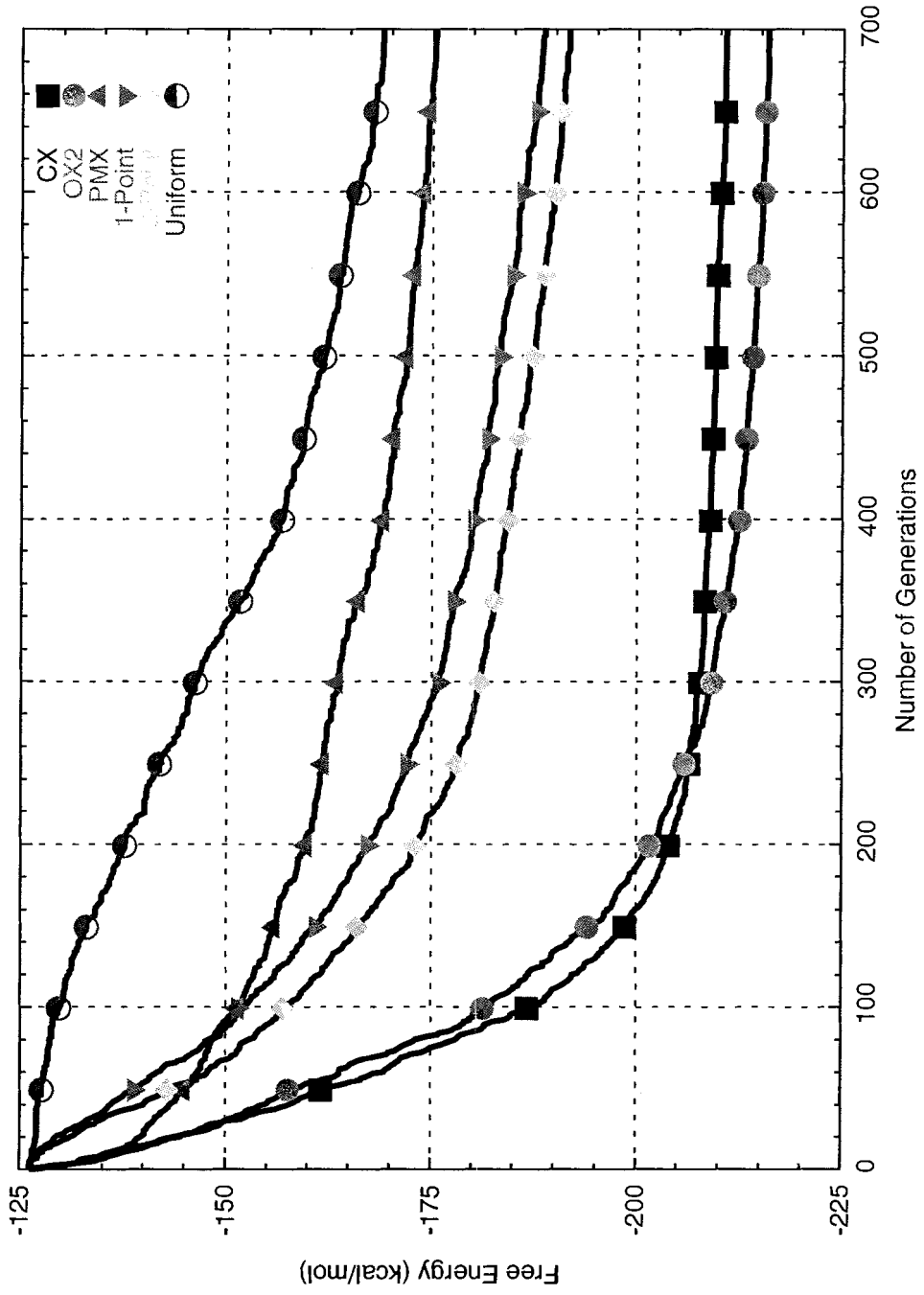

Figure 6.6: This graph compares the behavior of the top three permutation crossover operators and top three binary crossover operators with *Hildenbrandia rubra* using STDS. The graph follows the lowest energy structure from an average of 30 random seeds.

6.3.4 Selection

Lastly, selection strategies are compared with the top permutation crossover operators. Figure 6.7 shows the comparison of OX2, CX, and PMX with both STDS and KBR under the same conditions.

The graph shows that in the first 50 generations, KBR is very aggressive at finding low energy structures. However, KBR has the property of increased convergence velocity. This rapid convergence removes a lot of the diversity in the population, rendering it more difficult to make progress in subsequent generations.

The CX and OX2 experiment show that at approximately generation 150, STDS improves on their KBR counterparts. Maintaining diversity throughout the 700 generations allows these STDS experiments to improve by recombination of different permutations.

The PMX crossover operator, on the other hand, benefits greatly from the added selection pressure of KBR. Without KBR, PMX makes the slowest progress toward lower energy structures compared to all other experiments in this graph. However, with KBR enabled, PMX yields similar results to OX2 after 700 generations.

KBR is beneficial in the short term making fast progress toward lower energy structure in the early generations, but STDS is beneficial in the long run by maintaining diversity in the population for successful recombination. However, STDS seems to be only beneficial with more aggressive crossover operators such as CX and OX2.

6.3.5 Crossover and mutation rates

To determine the optimal parameter settings for RnaPredict, many runs were done using different selection strategies, crossover rates, and mutation rates using OX2, CX, and PMX crossover operators.

Table 6.2 shows all the parameters tested with the OX2 crossover operator. Since the convergence behaviour should not be affected by the thermodynamic model, all experiments were performed using INN-HB. In this table, the first column represents the free energy of the lowest energy structure found for a particular parameter setting. The second column shows the selection strategy, STDS or KBR. The third column shows the crossover operator used. The fourth and fifth columns shows the crossover and mutation rates, while the last column shows the thermodynamic model used.

The results from Table 6.2 show that STDS outperforms KBR when coupled with high

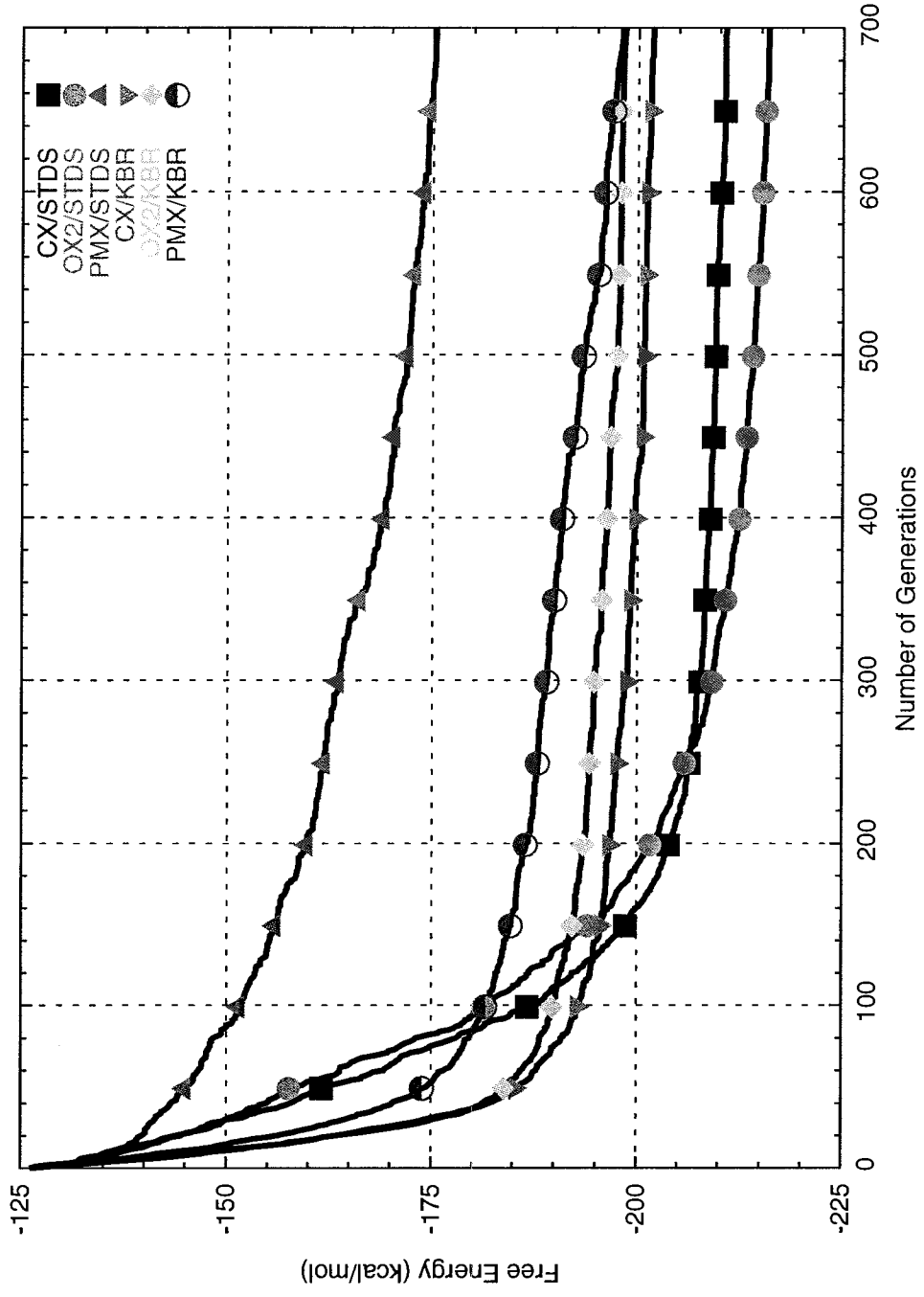

Figure 6.7: This graph compares the behavior of the top three permutation crossover operators with *Hildenbrandia rubra* using STDS and KBR. The graph follows the lowest energy structure from an average of 30 random seeds.

crossover and mutation rates. The next block of experiments shows that KBR also needs high crossover and mutation rates to be successful in finding low energy structures. Lastly, using low crossover rates with low mutation rates fails to perform well in finding low energy structures with this crossover operator. OX2 is a very aggressive crossover operator. High crossover rate is required to evolve the population to lower energy structures. Recombination with OX2 transmits a large number of genes maintaining absolute positions while maintaining relative positions for remaining genes. To ensure progress, a high mutation rate must be used to maintain diversity.

Table 6.3 shows similar behaviour with the CX crossover operator. Again, high crossover and mutation rates with STDS yield the best results followed by high rates with KBR. Again, low crossover and mutation rates perform poorly regardless of the choice between STDS and KBR.

The CX operator maintains absolute positions after recombination. A high rate of crossover is required to make progress to lower energy structures. Crossover allows to transmit good building blocks to the next generation. CX maintains absolute positions for many genes. In order to avoid too rapid a convergence in the population, the mutation rate must be high to increase diversity.

The third crossover operator tested was PMX and the results are shown in Table 6.4. This table shows a different trend. The results show that KBR experiments yield lower energy structures with high crossover and mutation rates. Even more interesting is that for high crossover and mutation rates with STDS, RnaPredict performs poorly yielding the highest energy structures.

PMX does not maintain absolute positions as OX2 and CX do. Because of this, PMX requires a high crossover and mutation rate but also requires KBR to increase selection pressure. KBR helps by avoiding the propagation of the worst individual by keeping the best parent and the best child after crossover.

The data in these three tables show that high crossover and mutation rates yield the best results in most cases. With these three operators, only PMX differs by yielding high energy structures with high crossover rates and high mutation rates with STDS. This was also found to be consistent across different sequences.

To make the results more manageable, a set of uniform parameter settings were chosen for all experiments. A crossover rate, P_c, of 0.7 and a mutation rate, P_m, of 0.8 is chosen for all experiments. These parameters adequately represent a high crossover rate and a high

Table 6.2: Parameter settings tested with *Homo Sapiens* sequence using OX2

ΔG (kcal/mol)	Selection	Crossover	Pc	Pm	Model
-269.88	STDS	OX2	0.6	0.8	INNHB
-269.77	STDS	OX2	0.6	0.9	INNHB
-269.74	STDS	OX2	0.6	0.6	INNHB
-268.87	STDS	OX2	0.8	0.6	INNHB
-268.83	STDS	OX2	0.6	0.2	INNHB
-268.52	STDS	OX2	0.8	0.9	INNHB
-268.24	STDS	OX2	0.8	0.2	INNHB
-267.83	STDS	OX2	0.8	0.8	INNHB
-267.08	STDS	OX2	0.8	0.4	INNHB
-266.98	STDS	OX2	0.6	0.4	INNHB
-264.25	STDS	OX2	0.4	0.9	INNHB
-263.78	STDS	OX2	0.4	0.6	INNHB
-262.12	STDS	OX2	0.4	0.4	INNHB
-261.33	STDS	OX2	0.4	0.8	INNHB
-260.91	KBR	OX2	0.8	0.8	INNHB
-260.61	STDS	OX2	0.4	0.2	INNHB
-259.57	KBR	OX2	0.8	0.9	INNHB
-259.41	KBR	OX2	0.8	0.6	INNHB
-257.34	KBR	OX2	0.6	0.9	INNHB
-256.21	KBR	OX2	0.8	0.4	INNHB
-255.92	KBR	OX2	0.4	0.9	INNHB
-255.43	KBR	OX2	0.6	0.8	INNHB
-253.84	KBR	OX2	0.6	0.6	INNHB
-253.70	KBR	OX2	0.4	0.8	INNHB
-253.44	KBR	OX2	0.2	0.9	INNHB
-252.15	KBR	OX2	0.8	0.2	INNHB
-250.21	KBR	OX2	0.6	0.4	INNHB
-250.12	KBR	OX2	0.4	0.6	INNHB
-249.39	KBR	OX2	0.4	0.4	INNHB
-248.68	KBR	OX2	0.2	0.6	INNHB
-247.65	KBR	OX2	0.2	0.8	INNHB
-246.96	STDS	OX2	0.2	0.6	INNHB
-245.89	KBR	OX2	0.6	0.2	INNHB
-245.60	KBR	OX2	0.2	0.4	INNHB
-245.29	STDS	OX2	0.2	0.8	INNHB
-244.30	KBR	OX2	0.4	0.2	INNHB
-244.20	STDS	OX2	0.2	0.9	INNHB
-241.81	STDS	OX2	0.2	0.4	INNHB
-238.93	KBR	OX2	0.2	0.2	INNHB
-236.70	STDS	OX2	0.2	0.2	INNHB

Table 6.3: Parameter settings tested with *Homo Sapiens* sequence using CX

ΔG (kcal/mol)	Selection	Crossover	Pc	Pm	Model
-268.34	STDS	CX	0.8	0.8	INNHB
-268.25	STDS	CX	0.8	0.6	INNHB
-267.80	STDS	CX	0.8	0.9	INNHB
-267.32	STDS	CX	0.8	0.4	INNHB
-267.08	STDS	CX	0.6	0.6	INNHB
-267.05	STDS	CX	0.6	0.8	INNHB
-266.99	STDS	CX	0.8	0.2	INNHB
-266.70	STDS	CX	0.6	0.4	INNHB
-265.74	STDS	CX	0.6	0.9	INNHB
-264.15	STDS	CX	0.6	0.2	INNHB
-258.87	KBR	CX	0.8	0.8	INNHB
-256.42	STDS	CX	0.4	0.8	INNHB
-254.87	KBR	CX	0.8	0.9	INNHB
-254.77	KBR	CX	0.8	0.6	INNHB
-254.10	KBR	CX	0.6	0.8	INNHB
-253.72	STDS	CX	0.4	0.9	INNHB
-253.07	KBR	CX	0.4	0.8	INNHB
-252.57	KBR	CX	0.6	0.9	INNHB
-252.12	KBR	CX	0.4	0.9	INNHB
-250.28	KBR	CX	0.8	0.4	INNHB
-250.07	STDS	CX	0.4	0.2	INNHB
-249.79	KBR	CX	0.4	0.6	INNHB
-249.58	KBR	CX	0.6	0.6	INNHB
-249.19	KBR	CX	0.2	0.9	INNHB
-249.09	STDS	CX	0.4	0.6	INNHB
-248.37	STDS	CX	0.4	0.4	INNHB
-248.18	KBR	CX	0.2	0.8	INNHB
-247.27	KBR	CX	0.6	0.4	INNHB
-245.87	KBR	CX	0.8	0.2	INNHB
-245.08	KBR	CX	0.2	0.6	INNHB
-244.78	KBR	CX	0.4	0.4	INNHB
-241.79	KBR	CX	0.4	0.2	INNHB
-241.49	KBR	CX	0.2	0.4	INNHB
-240.24	KBR	CX	0.6	0.2	INNHB
-239.08	STDS	CX	0.2	0.6	INNHB
-237.67	STDS	CX	0.2	0.9	INNHB
-236.22	KBR	CX	0.2	0.2	INNHB
-236.17	STDS	CX	0.2	0.8	INNHB
-233.48	STDS	CX	0.2	0.4	INNHB
-229.19	STDS	CX	0.2	0.2	INNHB

Table 6.4: Parameter settings tested with *Homo Sapiens* sequence using PMX

ΔG (kcal/mol)	Selection	Crossover	Pc	Pm	Model
-254.16	KBR	PMX	0.8	0.9	INNHB
-253.69	KBR	PMX	0.8	0.8	INNHB
-253.06	KBR	PMX	0.6	0.9	INNHB
-252.41	KBR	PMX	0.4	0.9	INNHB
-250.76	KBR	PMX	0.8	0.4	INNHB
-250.47	KBR	PMX	0.2	0.9	INNHB
-250.36	KBR	PMX	0.6	0.8	INNHB
-249.58	KBR	PMX	0.4	0.8	INNHB
-248.79	KBR	PMX	0.4	0.6	INNHB
-248.73	KBR	PMX	0.6	0.6	INNHB
-248.34	KBR	PMX	0.8	0.6	INNHB
-247.48	KBR	PMX	0.8	0.2	INNHB
-246.88	KBR	PMX	0.2	0.8	INNHB
-245.48	KBR	PMX	0.2	0.6	INNHB
-245.10	KBR	PMX	0.6	0.4	INNHB
-244.43	STDS	PMX	0.2	0.8	INNHB
-243.89	STDS	PMX	0.4	0.2	INNHB
-242.80	STDS	PMX	0.4	0.9	INNHB
-242.15	KBR	PMX	0.2	0.4	INNHB
-241.87	STDS	PMX	0.2	0.9	INNHB
-241.75	KBR	PMX	0.6	0.2	INNHB
-241.54	STDS	PMX	0.2	0.6	INNHB
-241.52	STDS	PMX	0.4	0.6	INNHB
-241.33	STDS	PMX	0.4	0.8	INNHB
-239.46	KBR	PMX	0.4	0.4	INNHB
-239.16	STDS	PMX	0.2	0.4	INNHB
-238.39	STDS	PMX	0.4	0.4	INNHB
-238.26	KBR	PMX	0.4	0.2	INNHB
-237.61	STDS	PMX	0.2	0.2	INNHB
-233.25	KBR	PMX	0.2	0.2	INNHB
-222.19	STDS	PMX	0.6	0.9	INNHB
-221.29	STDS	PMX	0.6	0.4	INNHB
-220.51	STDS	PMX	0.6	0.2	INNHB
-220.05	STDS	PMX	0.6	0.8	INNHB
-217.50	STDS	PMX	0.6	0.6	INNHB
-216.06	STDS	PMX	0.8	0.9	INNHB
-212.62	STDS	PMX	0.8	0.2	INNHB
-211.84	STDS	PMX	0.8	0.4	INNHB
-211.82	STDS	PMX	0.8	0.8	INNHB
-211.13	STDS	PMX	0.8	0.6	INNHB

mutation rate since no single parameter set was found to be best overall.

6.4 Pseudoknots

A simple pseudoknot is defined as two pairs of bases, (i, j) and (i', j') such that $i < i' < j < j'$. Figure 6.8 illustrates this mathematical definition.

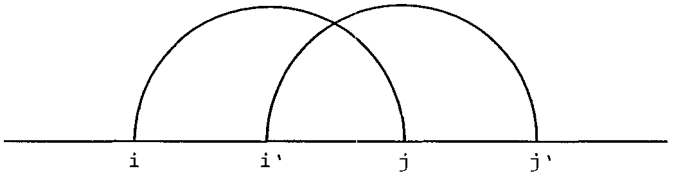

Figure 6.8: A graphical illustration of a simple pseudoknot.

Initially, the decoder would try to add as many helices as possible by reading the permutation from left to right or by adding all helices from the 1 bits in bit-strings. Adding helices without taking the resulting structure into account can add many pseudoknots. Pseudoknots are RNA sub-structures and an example is shown in Figure 6.9.

Too many pseudoknots were predicted with the original GA design and no thermodynamic penalties are currently implemented. Fortunately, pseudoknots occur infrequently in nature due to their low stability by comparison to regular stacked pairs.

To increase prediction accuracies, the formation of pseudoknots was disabled. This was done by disallowing pseudoknots to be constructed by the decoder which decodes the permutation/bit-string (genotype) to produce the final structure (phenotype).

6.5 Chapter summary

This chapter provides a rationale for the parameters chosen for the GA. Encoding, crossover operators and selection strategies were systematically varied. The results suggest that permutation encoding outperforms binary encoding. OX2 and CX coupled with STDS found lower energy structures more often than any other crossover operator and selection strategy combination. However, PMX requires to be coupled with KBR to increase the selection pressure. By systematically varying P_c and P_m, it was found that high rates were favorable.

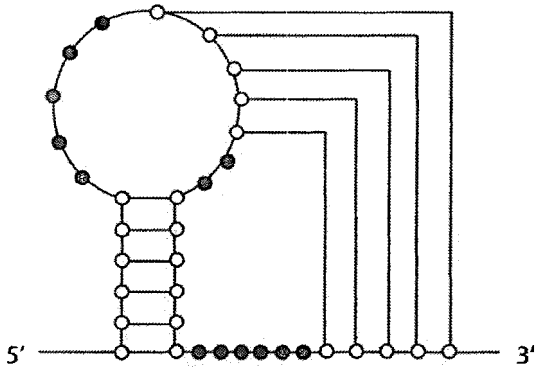

Figure 6.9: A diagram representing a pseudoknot. In this diagram, the hairpin loop has bases paired to it from a different part of the RNA sequence. Although pseudoknots do occur in real structures, their frequency is very low. Disallowing their formation improved the results dramatically. Figure taken from [3], page S323, permission granted by authors.

Without a proper model for pseudoknots, these were over-predicted by the GA. Disallowing pseudoknot formation in the decoder improved results. This decision may also allow a fairer comparison to other methods such as Nussinov and *mfold* DPAs since these algorithms cannot predict pseudoknots.

Chapter 7

Comparison to known structures

The quality of structure prediction is related to the similarity of the predicted structures to the known structures. The more "similar" the predicted structure is to the natural fold, the higher the accuracy. One particular quantitative metric to measure similarity counts the number of correctly predicted base pairs. The larger the number of base pairs correctly predicted, the higher the quality of the structure. However, a large number of correctly predicted base pairs does not ensure that all the substructures, such as hairpin loops, bulges, and internal loops, will be correctly predicted.

The results are presented here for five sequences. Tests were done using various parameters. The discussion is focused on the lowest energy structures found using CX, OX2, and PMX. The structures with the highest number of correctly predicted base pairs found with these parameter sets are also discussed. The generated structures will be compared to known structures. Assessing the quality of RnaPredict's results is done through quantitative measurement of the number of correctly predicted base pairs. A qualitative comparison is done by visually inspecting where predicted and known structures overlap, and determining which substructures are correctly predicted. False positive predictions are also considered. These are base pairs that are predicted but are not found in the known structure.

The parameters used are listed in Table 7.1.

7.1 *Xenopus laevis* - 945 nt

The longest sequence discussed in detail is a 945 nucleotide *Xenopus laevis* sequence.

Figure 7.1 shows a typical experiment for *Xenopus laevis*. The lighter outer envelope

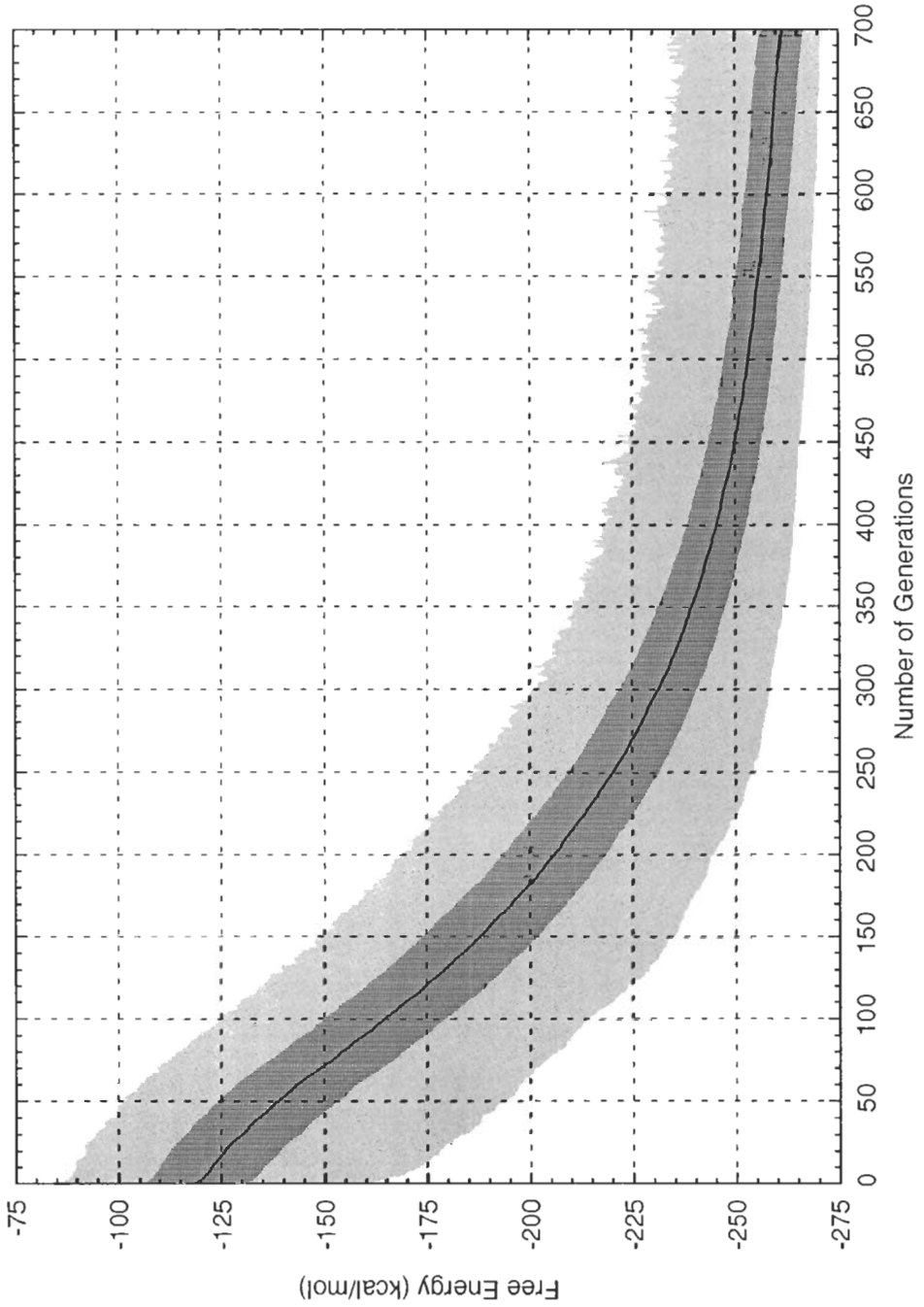

Figure 7.1: *Xenopus laevis*, $P_m = 0.8$, $P_c = 0.7$, population size = 700, CX, STDS, 1-elitism, average of 30 random seeds using INN-HB as the thermodynamic model. This experiment was able to correctly predict, on average, 23.0% of the base pairs from the known structure.

Table 7.1: Genetic algorithm parameters

Pop. Size	700
Generations	700
Crossover Operators	CX, OX2, PMX
P_c	0.7
P_m	0.8
Selection	STDS, KBR
Elitism	1
Thermodynamic Models	INN, INN-HB
Random seeds	30
Allow pseudoknots	No

of the plot represents the extremities of each generation (members with maximum and minimum energies). The darker inner envelope shows the mean free energy of the population with standard deviation. This particular graph is an experiment of 30 averaged runs for 700 generations. The graph, for the first 300 generations, shows a rapid decrease in the average free energy of the population. After this point, the graph's slope increases slowly but there is still a strong tendency for lower energy structures at each generation even after 700 generations.

Table 7.2: Results of comparison with known *Xenopus laevis* structure grouped by thermodynamic model. The known structure contains 251 base pairs. Each row represents an experiment consisting of 30 averaged runs.

ΔG (kcal / mol)	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-272.22	238.8	**62.9**	**25.0**	OX2	STDS	INNHB
-270.94	239.6	57.7	23.0	CX	STDS	INNHB
-254.31	233.2	49.8	19.9	OX2	KBR	INNHB
-253.17	232.2	48.7	19.4	CX	KBR	INNHB
-250.81	231.8	46.9	18.7	PMX	KBR	INNHB
-210.76	218.3	30.0	12.0	PMX	STDS	INNHB
-264.3	240.4	**62.9**	**25.1**	CX	STDS	INN
-261.7	239.9	61.4	24.5	OX2	STDS	INN
-249.1	236.1	49.0	19.5	OX2	KBR	INN
-247.7	233.6	42.9	17.1	CX	KBR	INN
-242.7	232.9	45.1	18.0	PMX	KBR	INN
-202.9	216.6	24.1	9.6	PMX	STDS	INN

Table 7.2 describes the results from RnaPredict after 700 generations. The first column shows the free energy of the average lowest energy structure from the given parameter set. In this case, it is an average of the lowest free energy structure for 30 random seeds after 700 generations within the given parameter set. The second column presents the average number of predicted base pairs from the lowest free energy structures. This average comes from 30 runs, each with a different random seed, after 700 generations. The third column lists the number of correctly predicted base pairs in the average predicted structure, while the fourth column shows the percentage of correctly predicted base pairs. The fifth column displays which crossover operator was used. The sixth column shows whether STDS or KBR was used, while the last column presents the chosen thermodynamic model. Each row represents a different crossover operator, selection strategy, or thermodynamic model. Since the free energy metric is incompatible between different thermodynamic models, the experiments have been grouped by the chosen model. Within each thermodynamic model group, the experiments are sorted by average minimum free energy structure after 700 generations. Each row with bold entries shows the parameter set that correctly predicted the highest number of pairs within the thermodynamic model.

To recap, Table 7.2 lists results from 360 individual runs. Each row consists of an experiments consisting of 30 averaged runs.

Table 7.2 shows that both INN-HB and INN perform similarly with INN being slightly better by finding more correct base pairs with its lowest energy structures. Within both thermodynamic models, the lowest energy structures were also the ones found to contain the highest number of known base pairs. With both models, CX and OX2 using STDS yielded the best results, finding lower energy structures more often than any other parameter set. The overall best experiment was found using the CX operator with STDS and INN. On average, this experiment predicted a structure with 25.1% of the known base pairs present. In second place was the OX2 experiment with STDS with INNHB predicting 25.0% of the correct base pairs on average.

Table 7.3 shows the single lowest free energy structure found with each crossover operator/thermodynamic model combination. The first column shows the lowest free energy structure found within the given parameter set. The second column shows the number of times a structure of same energy was found within the 30 seeds. The third column shows the generation number at which this structure was found. If the structure is found more than once, then the average generation number of all 30 seeds, in which this structure is found,

Table 7.3: Best results of comparison with known *Xenopus laevis* structure grouped by thermodynamic model. The known structure contains 251 base pairs. Best single run ranked by free energy.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-291.82	1	678	255	77	30.7	OX2	STDS	INNHB
-285.75	1	665	250	**82**	**32.7**	CX	STDS	INNHB
-282.23	1	572	245	72	28.7	OX2	KBR	INNHB
-272.95	1	523	253	57	22.7	CX	KBR	INNHB
-271.45	1	612	235	61	24.3	PMX	KBR	INNHB
-240.44	1	633	237	36	14.3	PMX	STDS	INNHB
-279.8	1	599	251	**75**	**29.9**	CX	STDS	INN
-277.3	1	618	253	75	29.9	OX2	STDS	INN
-267.1	1	516	249	51	20.3	CX	KBR	INN
-263.9	1	579	248	58	23.1	PMX	KBR	INN
-262.4	1	660	236	44	17.5	OX2	KBR	INN
-230.4	1	656	220	35	13.9	PMX	STDS	INN

is reported. The fourth column shows the number of predicted base pairs in the structure. The fifth lists the number of correctly predicted base pairs while the sixth column shows the percentage of known base pairs that were correctly predicted. The final three columns show the crossover operator, the selection strategy, and thermodynamic model, respectively.

The results show that the OX2 crossover operator was able to find the single lowest energy structure with STDS and INN-HB. This structure had a free energy of -291.82 kcal/mol and was found by a single random seed after 678 generations. The predicted structure contained 255 base pairs which overlapped with 30.7% of the known structure. However, a structure was found with INN-HB that contained an even higher number of base pairs, but had a slightly higher free energy. This structure was predicted with the CX crossover operator using STDS after 665 generations. The predicted structure contained 250 base pairs. The correct base pairs accounted for 32.7% of the known structure.

With the INN model, the single lowest energy structure was found with a single random seed after 599 generations. This run made use of the CX crossover operator and STDS. The structures energy was evaluated at -279.8 kcal/mol and contained 29.9% of the known base pairs. In this case, the best INN predicted structure was less accurate than the best

INN-HB predicted structure.

Every experiment finds a lowest energy structure after 700 generations from a single run. Most often, runs with different random seeds find different structures. Table 7.4 shows the structure with the highest number of correct base pairs regardless of free energy. This structure is still a low energy structure since it is the lowest energy structure found with RnaPredict after 700 generation using a particular single random seed. Out of all the structures predicted (12 experiments, 30 runs per experiment, total 360 runs) after 700 generations, a single run with the CX crossover operator, STDS and INN was found with more correct base pairs than any run listed above. This run predicted 37.1% of the known structure with a free energy of -267.8 kcal/mol. This predicted structure is a considerable improvement over the one with highest number of correctly predicted base pairs in Table 7.3. This structure is not listed in Table 7.3 because it is of higher energy than the lowest energy structure found with the same parameter set.

Table 7.4: Single run with highest number of correctly predicted base pairs of *Xenopus laevis*, regardless of free energy grouped by thermodynamic model. The known structure contains 251 base pairs.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-282.50	1	665	251	**91**	**36.3**	OX2	STDS	INNHB
-277.95	1	660	242	83	33.1	CX	STDS	INNHB
-273.11	1	632	239	85	33.9	OX2	KBR	INNHB
-269.88	1	639	236	83	33.1	CX	KBR	INNHB
-268.73	1	550	238	74	29.5	PMX	KBR	INNHB
-212.49	1	571	212	56	22.3	PMX	STDS	INNHB
-267.8	1	676	240	**93**	**37.1**	CX	STDS	INN
-264.8	1	566	245	84	33.5	OX2	STDS	INN
-262.1	1	685	255	77	30.7	PMX	KBR	INN
-259.7	1	376	242	73	29.1	OX2	KBR	INN
-249.7	1	681	236	61	24.3	CX	KBR	INN
-210.9	1	672	223	50	19.9	PMX	STDS	INN

Consistent with the results of Section 4.2, Table 7.2, 7.3, and 7.4 show that lower energy structures contained a higher number of correctly predicted base pairs. Also, consistent with results in Chapter 6, runs using OX2 and CX combined with STDS were able to predict

lower energy structures than any other parameter set. RnaPredict was able to predict as much as 37.1% of the known base pairs in the best case. This result is encouraging as it shows that RnaPredict functions well as a search engine. As is demonstrated in the next sections, in most cases, the improvements and overall quality of the results increases when shorter sequences are considered.

7.2 *Drosophila virilis* - 784 nt

Table 7.5 shows that with both INN-HB and INN, the crossover operator able to predict the lowest free energy structures on average was OX2. With STDS and INN-HB, the OX2 experiment predicted an average structure where 12.7% of the known base pairs were correctly predicted. The second best result in terms of lowest energy was found with the CX crossover operator using STDS. This experiment improved on the OX2 result by predicting a structure with 13.1% of the known base pairs.

With INN, the experiment finding the lowest free energy structure also used OX2 and STDS. This experiment was able to predict structures with 239.2 base pairs and correctly predicted 16.5% of the base pairs in the known structure on average. This result improves on the best result found with INN-HB. Better yet, an experiment using the CX crossover operator, was able to predict 18.8% of the known base pairs correctly from its average structure containing 239.4 base pairs, but with a slightly higher free energy.

Each experiment finds a single lowest energy structure as shown in Table 7.6. The overall lowest energy structure was found with one random seed with OX2 and STDS using INN-HB. This run only managed to correctly predict 4.7% of the known base pairs with a structure of -197.03 kcal/mol after 663 generations. However, a run using PMX and STDS was able to better this result by predicting a structure with 9.9% of the known base pairs after 667 generations. This new structure had a free energy of -158.55 kcal/mol. Even if this structure is significantly higher in free energy when evaluated by the INN-HB model, it is able to predict almost twice as many base pairs correctly than the overall lowest energy structure.

The overall lowest energy structure found with INN was from a run using the OX2 crossover operator with STDS after 672 generations. The structure found had a free energy of -173.4 kcal/mol and was able to predict 16.7% of the known base pairs. As with INN-HB, a structure with higher free energy turned out to be more accurate than the lowest energy

Table 7.5: Results of comparison with known *Drosophila virilis* structure grouped by thermodynamic model. The known structure contains 233 base pairs. Each row represents an experiment consisting of 30 averaged runs.

ΔG (kcal / mol)	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-177.75	239.5	29.6	12.7	OX2	STDS	INNHB
-175.77	241.3	**30.4**	**13.1**	CX	STDS	INNHB
-165.94	234.6	28.0	12.0	OX2	KBR	INNHB
-160.43	230.1	22.2	9.5	CX	KBR	INNHB
-157.40	231.2	21.9	9.4	PMX	KBR	INNHB
-138.89	221.6	18.1	7.8	PMX	STDS	INNHB
-159.53	239.2	38.4	16.5	OX2	STDS	INN
-157.68	239.4	**43.7**	**18.8**	CX	STDS	INN
-145.47	232.5	33.3	14.3	OX2	KBR	INN
-144.80	233.7	33.6	14.4	CX	KBR	INN
-139.90	230.4	28.9	12.4	PMX	KBR	INN
-120.79	222.7	21.8	9.4	PMX	STDS	INN

Table 7.6: Best results of comparison with known *Drosophila virilis* structure grouped by thermodynamic model. The known structure contains 233 base pairs. Best single run ranked by free energy.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-197.03	1	663	250	11	4.7	OX2	STDS	INNHB
-190.76	1	610	243	18	7.7	CX	STDS	INNHB
-188.80	1	224	242	17	7.3	OX2	KBR	INNHB
-175.36	1	555	226	19	8.2	CX	KBR	INNHB
-171.33	1	417	235	16	6.9	PMX	KBR	INNHB
-158.55	1	667	227	**23**	**9.9**	PMX	STDS	INNHB
-173.4	1	672	245	39	16.7	OX2	STDS	INN
-171.3	1	675	241	**58**	**24.9**	CX	STDS	INN
-165.9	1	639	244	34	14.6	CX	KBR	INN
-159.8	1	590	243	26	11.2	OX2	KBR	INN
-154.9	1	635	225	50	21.5	PMX	KBR	INN
-143.5	1	628	229	34	14.6	PMX	STDS	INN

structure. In this case a structure found with the CX crossover operator and STDS, after 675 generations, contained 24.9% of the known base pairs. This is a large improvement over the lowest energy structure found with INN as well as the most accurate structure found with INN-HB.

After verifying each of the thirty randomly seeded runs with RnaPredict, one structure per parameter set was found to contain more correct base pairs. These results are listed in Table 7.7. The structure with the highest number of correctly predicted base pairs was found with OX2, STDS, and INN-HB. It had a higher free energy than the overall lowest free energy structure found at -176.68 kcal/mol. However, as many as 27.9% of the known base pairs were found in this structure.

Table 7.7: Single run with highest number of correctly predicted base pairs of *Drosophila virilis*, regardless of free energy grouped by thermodynamic model. The known structure contains 233 base pairs.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-176.68	1	689	242	**65**	**27.9**	OX2	STDS	INNHB
-171.48	1	633	236	53	22.7	CX	STDS	INNHB
-170.02	1	446	233	52	22.3	OX2	KBR	INNHB
-162.68	1	634	238	49	21.0	CX	KBR	INNHB
-153.22	1	684	223	47	20.2	PMX	KBR	INNHB
-143.94	1	659	225	42	18.0	PMX	STDS	INNHB
-161.8	1	575	232	**62**	**26.6**	CX	STDS	INN
-158.8	1	696	247	60	25.8	OX2	STDS	INN
-152.5	2	576	238	61	26.2	OX2	KBR	INN
-144.4	1	643	230	**62**	**26.6**	CX	KBR	INN
-141.9	1	553	233	52	22.3	PMX	KBR	INN
-119.9	2	525	224	41	17.6	PMX	STDS	INN

The results for *Xenopus laevis* were better than with *Drosophila virilis* and as shown in Table 7.4, RnaPredict was able to predict 36.3% of the correct base pairs with OX2, STDS, and INN-HB. These results still indicate that the RnaPredict is a good search engine by finding low energy structures, but, it is possible that the *Drosophila virilis* may not be a good candidate sequence as compared to *Xenopus laevis*. For instance, Table 4.7 lists a correlation coefficient of -0.50 with INN. Thus, the poor prediction seems more related to

a poor correlation between free energy and the number of correct base pairs than a problem with the GA search engine.

7.3 *Hildenbrandia rubra* - 543 nt

Table 7.8 shows that the OX2 crossover operator with STDS was able to predict structures closest to the known structure on average using both thermodynamic models. With both models, the lowest energy structure was also the one with the highest number of predicted base pairs.

Table 7.8: Results of comparison with known *Hildenbrandia rubra* structure grouped by thermodynamic model. The known structure contains 138 base pairs. Each row represents an experiment consisting of 30 averaged runs.

ΔG (kcal / mol)	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-215.86	160.1	**48.4**	**35.1**	OX2	STDS	INNHB
-210.72	157.3	43.0	31.2	CX	STDS	INNHB
-201.84	156.2	39.4	28.5	CX	KBR	INNHB
-198.48	153.9	33.6	24.3	OX2	KBR	INNHB
-198.07	154.5	31.0	22.5	PMX	KBR	INNHB
-175.36	149.1	22.3	16.2	PMX	STDS	INNHB
-198.5	160.7	**47.1**	**34.1**	OX2	STDS	INN
-195.3	159.3	39.8	28.8	CX	STDS	INN
-183.5	155.3	30.4	22.0	OX2	KBR	INN
-182.8	154.9	30.1	21.8	PMX	KBR	INN
-181.7	155.2	29.4	21.3	CX	KBR	INN
-160.0	147.9	19.8	14.3	PMX	STDS	INN

With INN-HB, the average highest number of correctly predicted base pairs was 35.1%. Using the INN model, RnaPredict was able to correctly predict 34.1% of the known base pairs. With this sequence, INN-HB was able to slightly outperform the INN model by correctly predicting 1% more base pairs.

Table 7.9 shows the single lowest free energy structure found after running RnaPredict for a maximum of 700 generations. The overall lowest energy structure, with INN-HB, was found using the OX2 crossover operator and STDS after 658 generations with a single random seed. This structure had a free energy of -224.66 kcal/mol with 44.9% of the known base pairs correctly predicted.

Table 7.9: Best results of comparison with known *Hildenbrandia rubra* structure grouped by thermodynamic model. The known structure contains 138 base pairs. Best single run ranked by free energy.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-224.66	1	658	164	**62**	**44.9**	OX2	STDS	INNHB
-222.40	1	631	165	38	27.5	CX	STDS	INNHB
-216.76	1	684	161	52	37.7	CX	KBR	INNHB
-215.37	1	417	157	51	37.0	OX2	KBR	INNHB
-210.71	1	690	161	42	30.4	PMX	KBR	INNHB
-201.82	1	543	159	52	37.7	PMX	STDS	INNHB
-209.9	1	553	169	61	44.2	OX2	STDS	INN
-208.0	1	443	170	49	35.5	CX	STDS	INN
-201.6	1	678	165	42	30.4	PMX	KBR	INN
-200.9	1	416	162	**62**	**44.9**	CX	KBR	INN
-194.2	1	637	157	49	35.5	OX2	KBR	INN
-175.3	1	367	153	24	17.4	PMX	STDS	INN

With the INN thermodynamic model, the single lowest energy structure was again found using the OX2 crossover operator and STDS after 553 generations. The structure was evaluated to a free energy of -209.9 kcal/mol and correctly predicting 44.2% of the known structures' base pairs. However, the lowest energy structure found with the CX operator and KBR was 9 kcal/mol higher in energy but predicted 44.9% of the known base pairs correctly after running for 416 generations. This particular run slightly outperforms the lowest free energy structure found with the INN model.

RnaPredict was able to find 57.2% of the known base pairs with a single randomly seeded run using OX2, STDS, and the INN model as listed in Table 7.10. This structure contained a total of 161 base pairs and a free energy of -176.7 kcal/mol. This result shows a rather impressive improvement on the overall lowest free energy structure found in Table 7.9. This structure does not appear in Table 7.9 because it was found with a run using a different random seed than the one that found the lowest energy structure.

The results for this structure show excellent results where RnaPredict can predict more than half of the known base pairs correctly. This is an excellent improvements on the *Drosophila virilis* sequence. Lower energy structures for the most part consistently contained

Table 7.10: Single run with highest number of correctly predicted base pairs of *Hildenbrandia rubra*, regardless of free energy grouped by thermodynamic model. The known structure contains 138 base pairs.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-219.51	1	649	163	**71**	**51.4**	OX2	STDS	INNHB
-210.50	1	536	165	58	42.0	CX	STDS	INNHB
-209.40	1	615	153	66	47.8	CX	KBR	INNHB
-201.82	1	543	159	52	37.7	PMX	STDS	INNHB
-200.65	1	663	158	50	36.2	PMX	KBR	INNHB
-196.84	1	176	154	62	44.9	OX2	KBR	INNHB
-201.7	1	629	161	**79**	**57.2**	OX2	STDS	INN
-200.9	1	416	162	62	44.9	CX	KBR	INN
-200.1	1	420	158	73	52.9	CX	STDS	INN
-194.0	1	275	158	57	41.3	OX2	KBR	INN
-177.8	1	594	144	56	40.6	PMX	KBR	INN
-168.3	1	521	153	41	29.7	PMX	STDS	INN

more known base pairs with both INN-HB and INN thermodynamic models. This shows that RnaPredict with both stacking energy models works well as a search engine for low energy structures containing a large number of correct base pairs.

7.4 *Haloarcula marismortui* - 122 nt

Table 7.11 shows the average results for the *Haloarcula marismortui* sequence. With the INN-HB model, two crossover operators were able to find the same structures on average. The OX2 and PMX crossover operators, with STDS, were used in these experiments. These lowest energy structures were also the structures with the highest number of correctly predicted base pairs. The average structure was evaluated at a free energy of -54.94 kcal/mol and contained 42.1% of the known base pairs.

With INN, three crossover operators performed equally to the INN-HB experiments when using STDS. These structures with 42.1% accuracy were found with CX, OX2 and PMX. The structures found were evaluated with a free energy of -52.8 kcal/mol. These structures were also the ones with the highest number of correctly predicted base pairs with

42.1%.

Table 7.11: Results of comparison with known *Haloarcula marismortui* structure grouped by thermodynamic model. The known structure contains 38 base pairs. Each row represents an experiment consisting of 30 averaged runs.

ΔG (kcal / mol)	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-54.94	33.0	**16.0**	**42.1**	OX2	STDS	INNHB
-54.94	33.0	**16.0**	**42.1**	PMX	STDS	INNHB
-54.93	33.1	15.3	40.4	CX	STDS	INNHB
-54.92	33.3	14.3	37.7	OX2	KBR	INNHB
-54.91	33.7	12.7	33.3	CX	KBR	INNHB
-54.86	33.6	12.7	33.4	PMX	KBR	INNHB
-52.8	30.0	**16.0**	**42.1**	CX	STDS	INN
-52.8	30.0	**16.0**	**42.1**	OX2	STDS	INN
-52.8	30.0	**16.0**	**42.1**	PMX	STDS	INN
-52.7	30.0	15.3	40.4	OX2	KBR	INN
-52.7	30.5	15.0	39.5	PMX	KBR	INN
-52.5	31.0	14.0	36.8	CX	KBR	INN

With this sequence, INN-HB and INN performed equally well on average. Table 7.12 shows the lowest free energy structures found with each parameter set. The results show that each and every parameter set was able to find the same minimal free energy structure within its respective thermodynamic model. With INN-HB, the structure found had a free energy of -54.94 kcal/mol. This structure contained 42.1% of the known base pairs. Although each run was able to find this structure, only OX2 and PMX, both using STDS, found it with all 30 random seeds. With OX2, the structure was found within 27.8 generations, on average. CX (STDS), OX2 (KBR), CX (KBR) and PMX (KBR) found the structure 28, 25, 19, and 18 times, respectively.

With INN, the lowest free energy structure found was -52.8 kcal/mol. This structure also contained 42.1% of the known base pairs. This structure was found with all 30 random seeds for the three crossover operators with STDS. The structure was found with as few as 26.6 generations on average with OX2. With KBR, OX2, PMX, and CX found the structure 28, 27, and 24 times, respectively.

Most runs found the same structure within each thermodynamic model, but there were a few runs that found structures with higher free energy. One of these was found using a single random seed with PMX, KBR, and INN-HB. It contained more correct base pairs

Table 7.12: Best results of comparison with known *Haloarcula marismortui* structure grouped by thermodynamic model. The known structure contains 38 base pairs. Best single run ranked by free energy.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-54.94	30	27.8	33	**16**	**42.1**	OX2	STDS	INNHB
-54.94	30	76.6	33	**16**	**42.1**	PMX	STDS	INNHB
-54.94	28	35.5	33	**16**	**42.1**	CX	STDS	INNHB
-54.94	25	83.7	33	**16**	**42.1**	OX2	KBR	INNHB
-54.94	19	75.6	33	**16**	**42.1**	CX	KBR	INNHB
-54.94	18	66.3	33	**16**	**42.1**	PMX	KBR	INNHB
-52.8	30	26.6	30	**16**	**42.1**	OX2	STDS	INN
-52.8	30	45.5	30	**16**	**42.1**	PMX	STDS	INN
-52.8	30	53.3	30	**16**	**42.1**	CX	STDS	INN
-52.8	28	87.8	30	**16**	**42.1**	OX2	KBR	INN
-52.8	27	73.1	30	**16**	**42.1**	PMX	KBR	INN
-52.8	24	98.9	30	**16**	**42.1**	CX	KBR	INN

than any structure listed in Table 7.12. This structure, highlighted in Table 7.13, had a free energy of -53.51 kcal/mol and contained 71.1% of the known base pairs. This predicted structure is a dramatic improvment with only a slightly higher free energy than the lowest energy structure found with INN-HB.

7.4.1 Graphical comparison

Comparing structures using quantitative measures, such as the number of correctly predicted base pairs, is useful, but comparing the overlap qualitatively can strengthen the interpretation of the results. Particularly, a qualitative graphical comparison can identify regions of high structural similarity between two structures, even if the quantitative overlap of base pairs in those regions is low. This happens in some cases such as when there is a shift in base pairs caused by a bulge that may be present in one structure and absent in the other.

Figure 7.2 shows a typical comparison between two structures of the same sequence. The figure shows how two *Haloarcula marismortui* structures overlap. The known structure is represented by the light grey bonds while the predicted structure is represented by dark

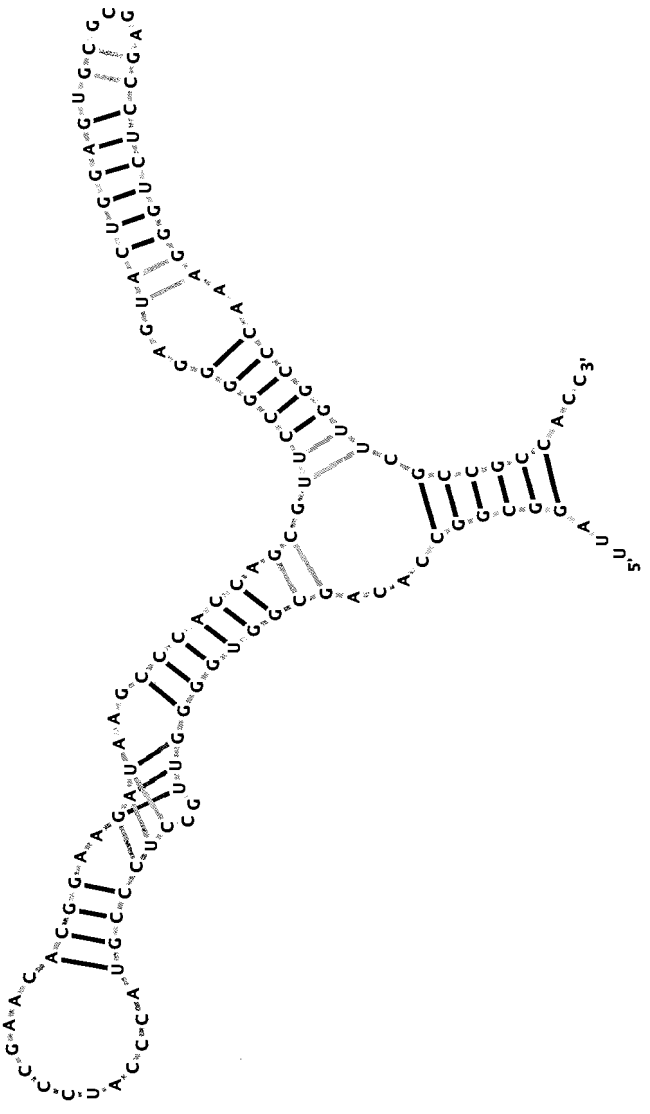

Figure 7.2: This is the overall single best structure found with RnaPredict without allowing pseudoknots for the 122 nucleotide sequence of *Haloarcula marismortui*. The known structure is depicted by the light grey bonds, the predicted structure is shown by the dark grey bond, while the overlap is shown by the black bonds. The predicted structure consists of six helices. This was found with a single PMX random seed using KBR and INN-HB.

Table 7.13: Single run with highest number of correctly predicted base pairs of *Haloarcula marismortui*, regardless of free energy grouped by thermodynamic model. The known structure contains 38 base pairs.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-54.94	30	27.8	33	16	42.1	OX2	STDS	INNHB
-54.94	30	76.6	33	16	42.1	PMX	STDS	INNHB
-54.94	28	35.5	33	16	42.1	CX	STDS	INNHB
-54.94	20	75.6	33	16	42.1	CX	KBR	INNHB
-54.94	25	83.7	33	16	42.1	OX2	KBR	INNHB
-53.51	1	20.0	30	**27**	**71.1**	PMX	KBR	INNHB
-52.8	30	26.6	30	**16**	**42.1**	OX2	STDS	INN
-52.8	30	45.5	30	**16**	**42.1**	PMX	STDS	INN
-52.8	30	53.3	30	**16**	**42.1**	CX	STDS	INN
-52.8	27	73.1	30	**16**	**42.1**	PMX	KBR	INN
-52.8	28	87.8	30	**16**	**42.1**	OX2	KBR	INN
-52.8	24	98.9	30	**16**	**42.1**	CX	KBR	INN

grey bonds. Where the known and the predicted structures overlap, the bonds are colored black.

Figure 7.2 shows the known structure and the overall best single structure generated by RnaPredict. This particular structure appears in Table 7.13 and was found with a single PMX run using KBR but was not the lowest energy structure. The lowest energy structure listed in Table 7.12 only contained 42.1% of the known base pairs but this new structure was of higher free energy. The agreement between the predicted and known structures is high with base pair overlap of 71.1%. Looking at the known structure more closely, interesting features are noted. First, the known structure contains non-canonical base pairs. The branch on the right shows two adjacent UU base pairs at its base. In the middle of the same branch, there is a GA pair. Also, near the middle section of the branch on the left, there is a CU pair. Since these pairs cannot be predicted by the helix generation model, they could be removed for the sake of a fairer comparison with the predicted structure. Removing these three pairs changes the structure considerably. The helix generation model does not allow helices shorter than three adjacent base pairs. With this in mind, the model could not predict the two base pairs forming the hairpin loop in the rightmost branch. With the same

reasoning, the removal of the GA pair in the middle of the same branch would not allow the adjacent AU pair to form. The base pairs AU and CG adjacent to the non-canonical CU pair would also not form. Also, at the root of the left branch, there is a bulge that could not be predicted since the GC stack at the base of the branch could not form with only a length of two.

Removing these eleven base pairs, which could not have been predicted by RnaPredict, from the known structure, leaves a total of 27 base pairs, which were all correctly predicted.

At a higher level, many similarities between the known structure (Figure 7.3) and the best predicted structure (Figure 7.4) are found. Both structures are composed of two hairpin loops, three internal loops, two dangling ends, one internal loop with three branches, and two internal loops with two branches. These findings show that RnaPredict is successful in correctly predicting the secondary structure of RNA molecules.

7.5 *Saccharomyces cerevisiae* - 118 nt

Table 7.14 shows the average results with the shortest sequence. The results show that with INN-HB, all parameter settings yielded the same result. The structure found had a free energy of -57.52 kcal/mol. This structure contained 89.2% of the known base pairs. With INN, a different structure was predicted with a free energy of -52.9 kcal/mol and contained 75.7% of the known base pairs.

Table 7.15 shows that all 30 seeds from each parameter setting were able to find the lowest energy structure within their respective thermodynamic model. Hence, the absolute lowest free energy structure found was the same as the aforementioned averaged free energy structure from Table 7.14. Since all 30 seeds for each parameter set found the same structure, Table 7.15 and Table 7.16 are equivalent.

7.5.1 Graphical omparison

As in the case of *Haloarcula marismortui*, interesting conclusions can be made by looking at qualitative, graphical comparisons between the most accurate predicted structure and the known structure.

The comparison between the structure with the highest number of correct base pairs and the known structure is seen in Figure 7.5. This figure shows that RnaPredict is able to predict as many as 89.2% of the known base pairs correctly but also correctly predicts

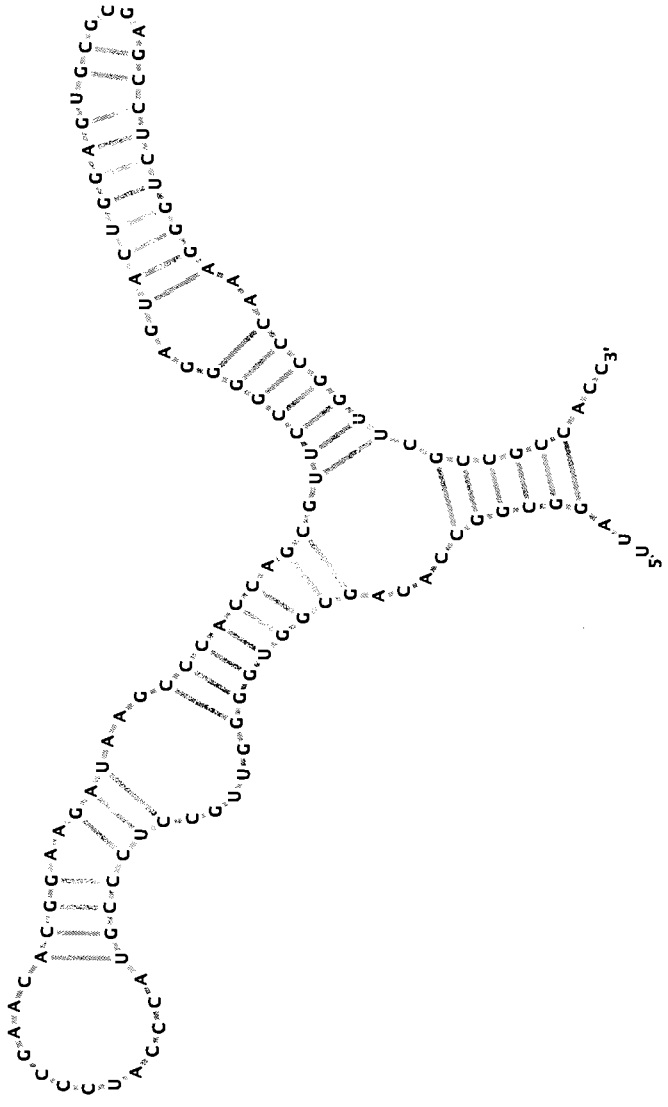

Figure 7.3: This is the known structure of *Haloarcula marismortui*. The base pairs are depicted by the light grey bonds. The structure consists of six helices.

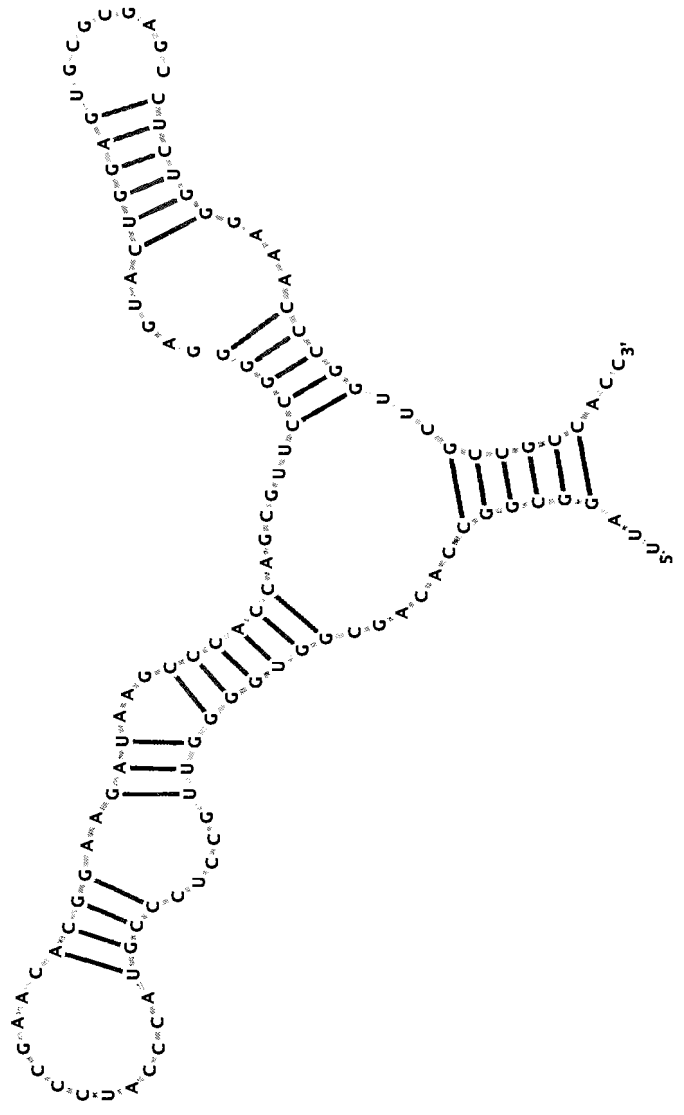

Figure 7.4: This is the overall single best structure found with RnaPredict without allowing pseudoknots for the 122 nucleotide sequence of *Haloarcula marismortui*. The predicted structure is shown by the dark grey bonds. The structure consists of six helices. This was found with a single PMX random seed using KBR and INN-HB.

Table 7.14: Results of comparison with known *Saccharomyces cerevisiae* structure grouped by thermodynamic model. The known structure contains 37 base pairs. Each row represents an experiment consisting of 30 averaged runs.

ΔG (kcal / mol)	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-57.52	39	**33**	**89.2**	CX	KBR	INNHB
-57.52	39	**33**	**89.2**	OX2	KBR	INNHB
-57.52	39	**33**	**89.2**	PMX	KBR	INNHB
-57.52	39	**33**	**89.2**	CX	STDS	INNHB
-57.52	39	**33**	**89.2**	OX2	STDS	INNHB
-57.52	39	**33**	**89.2**	PMX	STDS	INNHB
-52.9	40	**33**	**75.7**	OX2	KBR	INN
-52.9	40	**33**	**75.7**	PMX	KBR	INN
-52.9	40	**33**	**75.7**	CX	KBR	INN
-52.9	40	**33**	**75.7**	CX	STDS	INN
-52.9	40	**33**	**75.7**	OX2	STDS	INN
-52.9	40	**33**	**75.7**	PMX	STDS	INN

Table 7.15: Best results of comparison with known *Saccharomyces cerevisiae* structure grouped by thermodynamic model. The known structure contains 37 base pairs. Best single run ranked by free energy.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-57.52	30	4.7	39	**33**	**89.2**	OX2	KBR	INNHB
-57.52	30	5.0	39	**33**	**89.2**	CX	KBR	INNHB
-57.52	30	6.1	39	**33**	**89.2**	CX	STDS	INNHB
-57.52	30	6.6	39	**33**	**89.2**	OX2	STDS	INNHB
-57.52	30	8.4	39	**33**	**89.2**	PMX	KBR	INNHB
-57.52	30	9.2	39	**33**	**89.2**	PMX	STDS	INNHB
-52.9	30	5.7	40	**33**	**75.7**	OX2	KBR	INN
-52.9	30	6.8	40	**33**	**75.7**	OX2	STDS	INN
-52.9	30	7.0	40	**33**	**75.7**	CX	STDS	INN
-52.9	30	10.2	40	**33**	**75.7**	CX	KBR	INN
-52.9	30	11.2	40	**33**	**75.7**	PMX	STDS	INN
-52.9	30	15.4	40	**33**	**75.7**	PMX	KBR	INN

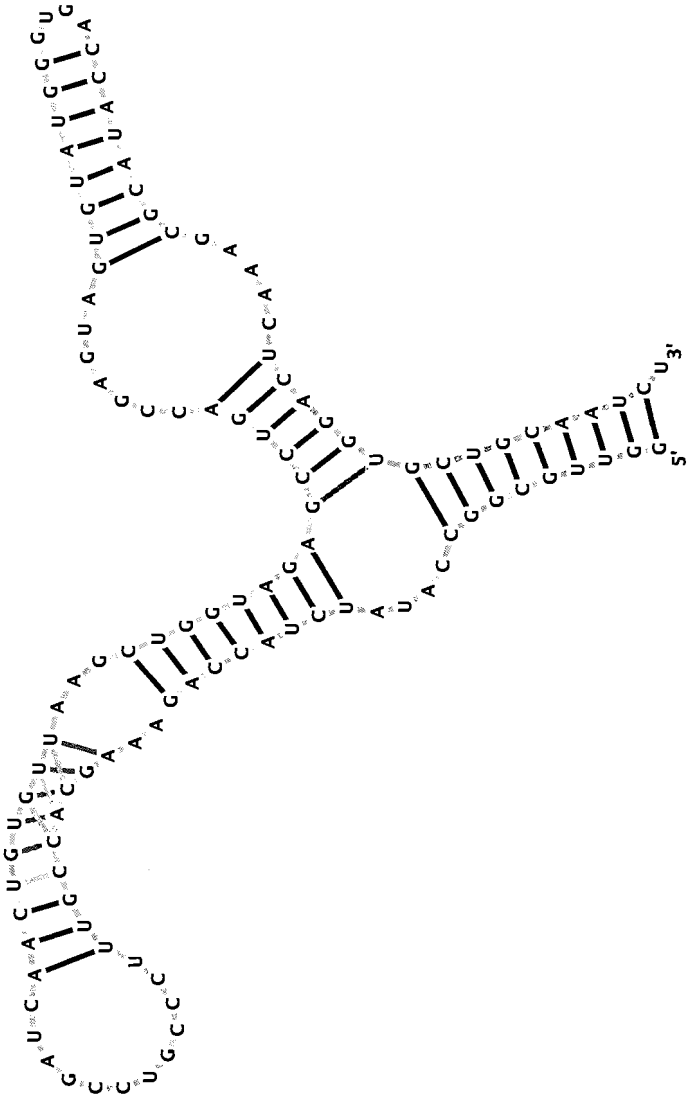

Figure 7.5: The above shows a comparison between the known and the highest number of correctly predicted base pairs using RnaPredict. The predicted base pairs are coloured in dark grey, the known are coloured in light grey, and the overlap is coloured in black. RnaPredict was able to predict 89.2% of the known *Saccharomyces cerevisiae* base pairs.

Table 7.16: Single run with highest number of correctly predicted base pairs of *Saccharomyces cerevisiae*, regardless of free energy grouped by thermodynamic model. The known structure contains 37 base pairs.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-57.52	30	4.7	39	**33**	**89.2**	OX2	KBR	INNHB
-57.52	30	5.0	39	**33**	**89.2**	CX	KBR	INNHB
-57.52	30	6.1	39	**33**	**89.2**	CX	STDS	INNHB
-57.52	30	6.6	39	**33**	**89.2**	OX2	STDS	INNHB
-57.52	30	8.4	39	**33**	**89.2**	PMX	KBR	INNHB
-57.52	30	9.2	39	**33**	**89.2**	PMX	STDS	INNHB
-52.9	30	5.7	40	**28**	**75.7**	OX2	KBR	INN
-52.9	30	6.8	40	**28**	**75.7**	OX2	STDS	INN
-52.9	30	7.0	40	**28**	**75.7**	CX	STDS	INN
-52.9	30	10.2	40	**28**	**75.7**	CX	KBR	INN
-52.9	30	11.2	40	**28**	**75.7**	PMX	STDS	INN
-52.9	30	15.4	40	**28**	**75.7**	PMX	KBR	INN

most of the known structure's substructures correctly. The known structure contains three branches and two internal loops. The predicted structure contains all these substructures with minor modifications due to the constraints in the helix generation algorithm.

RnaPredict's helix generation algorithm is bound by three rules: a helix must contain at least three stacked pairs, a helix must be connected by at least three nucleotides, and base pairs must be composed of GC, AU, or GU. Looking at the known structure in Figure 7.5 shows that there are two base pairs that the model cannot predict. The branch on the left hand side shows two CU base pairs in the known structure. For a fairer comparison, these could be removed. Removing the first CU base pair also removes the ability of the CU-AU-CG stack to form since the model does not allow for a helix of length two. If this helix, along with the second CU pair, is removed, RnaPredict has effectively correctly predicted 100% of the known base pairs. This lends further support to the notion that the GA search engine is very effective in identifying real structural elements, but is limited by the current helix generation model.

To aid in visually comparing the structures, the known (Figure 7.6) and the best structure (Figure 7.7) are also presented individually.

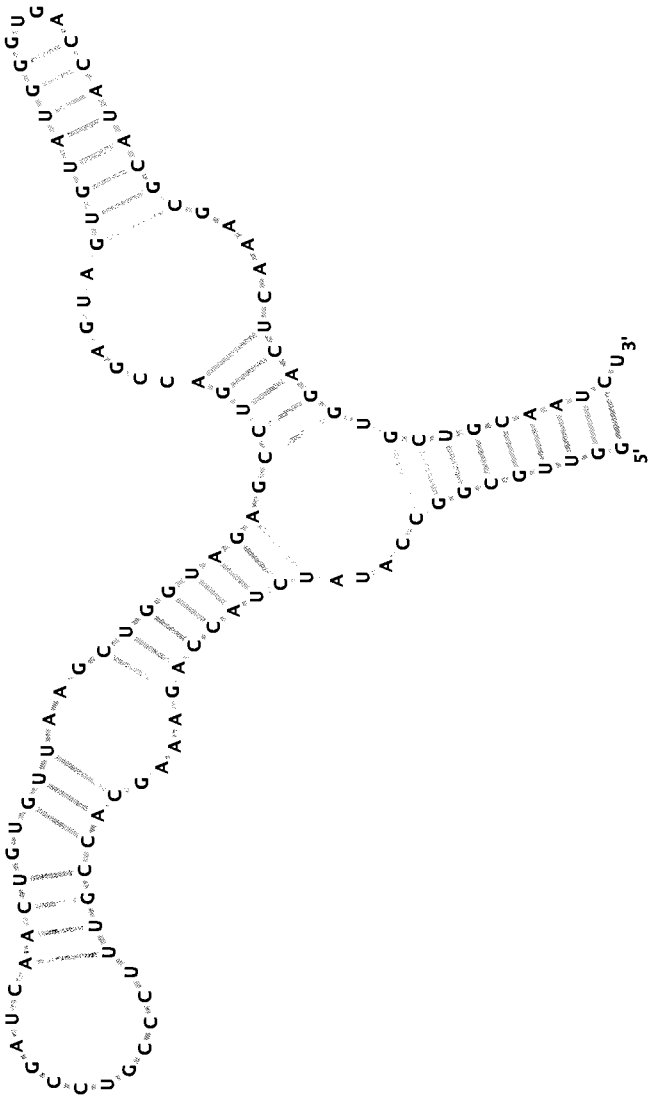

Figure 7.6: This is the known structure of *Saccharomyces cerevisiae*. The known base pairs are shown in light grey bonds.

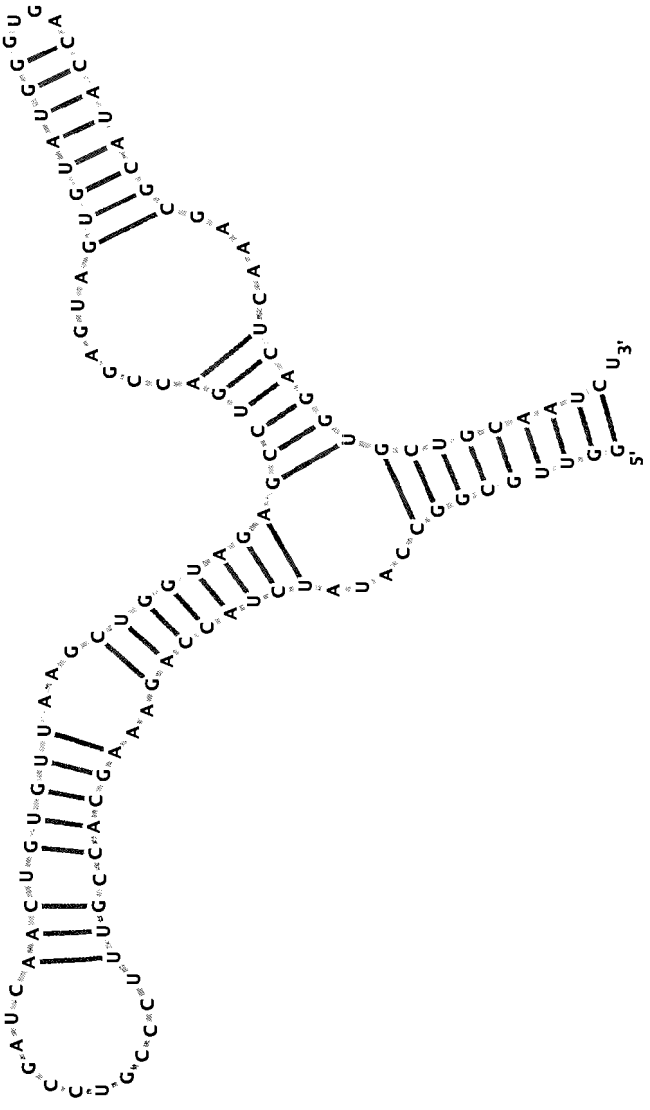

Figure 7.7: The above shows the structure with the highest number of correct base pairs with *Saccharomyces cerevisiae*. The dark grey base pairs correspond to the predicted structure.

The results for *Saccharomyces cerevisiae* are better than for any other sequence, including *Haloarcula marismortui* which is of similar length. One of the reason for this is the small number of non-canonical base pairs in *Saccharomyces cerevisiae*. Also, this sequence has very few base pairs that cannot be predicted by the current helix generation model.

7.6 Chapter summary

In this chapter, RnaPredict has shown its potential in finding low energy structure within a relatively small number of generations. For the most part, low energy structures contain more base pairs than high energy structures giving credibility to the thermodynamic models. With both INN-HB and INN, low energy structures contained a high number of base pairs. It was also found that, for the most part, the shorter the sequence, the higher the percentage of correctly predicted base pairs found. This can be attributed to the fact that short sequences have a smaller search space. Also, longer sequences may have required more generations to achieve better convergence. Another reason for the trend is that the thermodynamic parameters themselves were generated using short duplexes and may have a bias for short sequences with no provision for long range interactions.

For all cases, RnaPredict was able to improve on a random population by reducing the free energy of the population and increasing the number of correct base pairs making it an effective search engine.

Data for additional sequences can be found in Appendix A.

Chapter 8

Comparison to the Nussinov DPA

The Nussinov DPA [4] attempts to find structures with the maximum number of possible base pairs. The algorithm works by recursively calculating the optimal structure by finding the maximum number of base pairs in subsequences until the complete structure contains the largest number of base pairs.

Normally, the Nussinov DPA tries to maximize the number of base pairs in a structure regardless of the type. A simple modification [2] was added to the DPA allowing to set weights corresponding to the relative free energy or the relative number of hydrogen bonds in the base pairs.

In this research, the values were set to emulate the Major model from Section 3.1.1 by setting the weights for GC:AU:GU to 3:2:1. These weights reflect the relative stability of the base pairs.

A second set of weights were used to emulate the Mathews control model and is described in Section 3.1.2. In this model, the weights were set to 3:2:2 corresponding to the number of hydrogen bonds in GC, AU, and GU.

8.1 *Xenopus laevis* - 945 nt

Table 8.1 shows the results when using the Nussinov DPA. The results show that the maximum number of possible base pairs with this sequence within a single structure is 341. This structure contains 12.0% of the known base pairs. Changing the weights to be proportional to the number of hydrogen bonds in each base pair, 3:2:2, for GC, AU, and GU, respectively, the algorithm predicts a structure with 339 base pairs which contain 15.6% of the

real structure. Lastly, changing the weights to 3:2:1 to approximate the stability of the base pairs of GC, AU, and GU, respectively, predicted a structure with 333 base pairs where 18.7% of the real structure was correctly predicted.

Table 8.1: *Xenopus laevis*, Nussinov results. Number of known base pairs is 251.

GC:AU:GU Weights	Predicted BP	Correctly Predicted BP	Correctly Predicted (%)
1:1:1	341	30	12.0
3:2:2	339	39	15.6
3:2:1	333	47	18.7

Recalling RnaPredict's results from Table 7.2 shows that RnaPredict's average lowest energy structure found with OX2 and STDS contained 240.4 base pairs. This structure, found with INN, contained 25.1% of the known base pairs making this result far better than the best Nussinov result. Another structure found with INN-HB was almost as accurate with 238.8 base pairs containing 25.0% of those in the known structure. Both these structures contain far more correctly predicted base pairs than any Nussinov results.

8.2 *Drosophila virilis* - 784 nt

Table 8.2 shows the results when using the Nussinov DPA. The results show that the maximum number of possible base pairs with this sequence within a single structure is 320. This structure contains 12.4% of the known base pairs. Changing the weights to 3:2:2, for GC, AU, and GU, respectively, the algorithm predicts a structure with 319 base pairs that contain 9.9% of the natural fold. Lastly, changing the weights to 3:2:1 for GC, AU, and GU, respectively, predicted a structure with 309 base pairs where 9.0% of the real structure was correctly predicted.

Looking back at Table 7.5 shows that with INN-HB, the crossover operator able to predict the lowest free energy structures on average was OX2 with STDS. This experiment predicted an average structure containing 239.5 base pairs where 12.7% of the base pairs were correctly predicted. This result is better than the best result found with the Nussinov algorithm which correctly predicted 12.4% of the known structure. Moreover, the Nussinov DPA is prone to overprediction of base pairs (i.e.: it predicts many false positive base pairs

Table 8.2: *Drosophila virilis*, Nussinov results. Number of known base pairs is 233.

GC:AU:GU Weights	Predicted BP	Correctly Predicted BP	Correctly Predicted (%)
1:1:1	320	29	12.4
3:2:2	319	23	9.9
3:2:1	309	21	9.0

that are not found in the natural fold). Another experiment with CX and STDS was able to improve on this result predicting a structure with 13.1% of the known base pairs with a structure only containing 241.3 base pairs on average.

With INN, the experiment finding the lowest free energy structure was also found with OX2. This experiment was able to predict structures with 239.2 base pairs and correctly predicted 16.5% of the base pairs in the known structure on average. This result improves on the best result found with INN-HB. It also predicts more base pairs correctly than any Nussinov result. Better yet, an experiment using the CX crossover operator, was able to predict 18.8% of the known base pairs correctly from its average structure containing 239.4 base pairs, but with a slightly higher free energy.

8.3 *Hildenbrandia rubra* - 543 nt

The Nussinov results shown in Table 8.3 give us the upper bound on the number of base pairs possible in this sequence at 213 base pairs. This structure contains 5.0% of those found in the real structure. Changing the weights to be proportional to the number of hydrogen bonds in each base pair predicts a structure with 211 base pairs where 22.5% of the base pairs in the real structure are correctly predicted. Changing the weight for GU pairs to 1, because of its weaker stability, reduces the number of predicted base pairs to 205 but is still contains 22.5% of the known base pairs.

Recalling results from Table 7.8 show that the lowest energy structure contained, on average, 160.1 base pairs and 35.1% of the known base pairs. This structure contained far more correctly predicted base pairs making this result far superior to any Nussinov prediction.

Table 8.3: *Hildenbrandia rubra*, Nussinov results. Number of known base pairs is 138.

GC:AU:GU Weights	Predicted BP	Correctly Predicted BP	Correctly Predicted (%)
1:1:1	213	7	5.0
3:2:2	211	31	22.5
3:2:1	205	31	22.5

8.4 *Haloarcula marismortui* - 122 nt

Table 8.4 shows the results of the Nussinov DPA with *Haloarcula marismortui*. The results show that the maximum number of base pairs possible with this sequence is 45 base pairs. This structure contained 21.1% of the known base pairs. Changing the weights to 3:2:2, for GC, AU, and GU, respectively, yielded a less accurate prediction with a structure containing 44 base pairs with 10.5% of the known structure correctly predicted. Lastly, changing the weights to 3:2:1 for GC, AU, and GU, respectively gave the same result as in the 3:2:2 case.

Table 8.4: *Haloarcula marismortui*, Nussinov results. Number of known base pairs is 38.

GC:AU:GU Weights	Predicted BP	Correctly Predicted BP	Correctly Predicted (%)
1:1:1	45	8	21.1
3:2:2	44	4	10.5
3:2:1	44	4	10.5

Looking at the average results from Table 7.11 shows that the lowest energy structures were found with the OX2 and PMX crossover operators with STDS and INN-HB. These predicted structures contained 33 base pairs and 42.1% of the known base pairs. CX, OX2 and PMX performed equally well with INN and STDS but found a structure containing fewer base pairs, 30. This reduces the number of false positive predictions. In both INN-HB and INN, RnaPredict outperforms the Nussinov DPA and the number of false positives found with RnaPredict is far less than the number found with any Nussinov results.

8.5 *Saccharomyces cerevisiae* - 118 nt

Table 8.5 shows the results when using the Nussinov DPA. The results show that the maximum number of base pairs possible with this sequence is 45. Base pair maximization alone predicts 75.7% of the known base pairs correctly. This structure is shown in Figure 8.1. Changing the weights to be proportional to the number of hydrogen bonds in each base pair, 3:2:2, for GC, AU, and GU, respectively, the algorithm predicts a different structure also containing 45 base pairs which coincides with 75.7% of the base pairs in the real structure. Lastly, changing the weights to 3:2:1 to approximate the stability of the base pairs of GC, AU, and GU, respectively, predicted a structure with 44 base pairs with only 24.3% correct.

Table 8.5: *Saccharomyces cerevisiae*, Nussinov results. Number of known base pairs is 37.

GC:AU:GU Weights	**Predicted BP**	**Correctly Predicted BP**	**Correctly Predicted (%)**
1:1:1	45	28	75.7
3:2:2	45	28	75.7
3:2:1	44	9	24.3

Table 7.14 demonstrates that, with INN-HB, all experiments of RnaPredict were able to predict a structure that contained 39 base pairs and correctly predicted 89.2% of the known base pairs.

To give further evidence of the quality of RnaPredict's prediction, it can be compared to the structure generated by the Nussinov DPA. The structure generated by the Nussinov DPA is shown in Figure 8.1. This structure contained 75.7% of the known structure's base pairs.

A second diagram (Figure 8.2) shows which base pairs from Nussinov prediction, using base pair maximization, overlapped with the known structure. Figure 8.2 shows only the predicted structure (light grey) and highlights the bonds that overlap (black) to ease interpretation. This diagram should be compared to Figure 7.5 where 89.2% of the base pairs were correctly predicted. Although the number of base pairs correctly predicted is high for both prediction methods, there are very obvious differences between the two structures. The predicted Nussinov DPA structure in Figure 8.1 contains three major branches similar

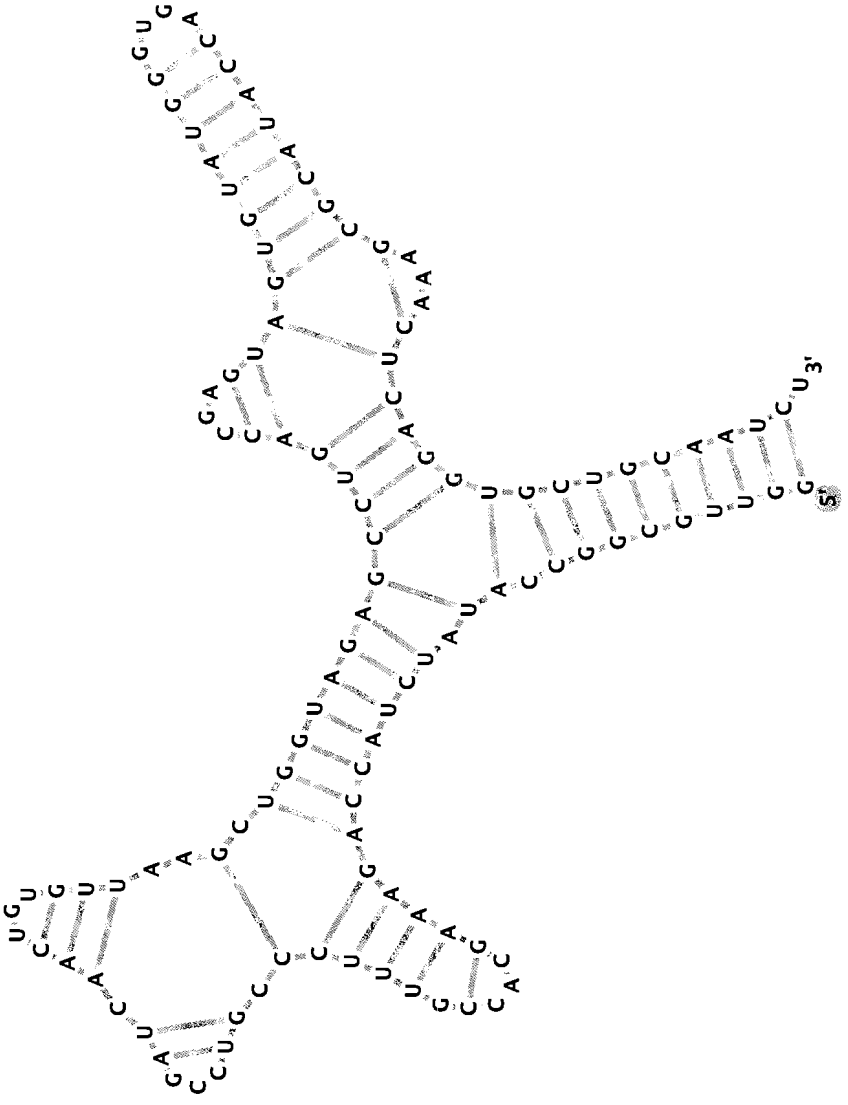

Figure 8.1: The above shows the structure predicted with the Nussinov DPA base pair maximization (1:1:1). The light grey base pairs correspond to the predicted structure. In this case, the Nussinov algorithm was able to predict 75.7% of the known base pairs.

to the real structure. However, upon closer inspection, the predicted structure fails to recognize the multi-branch loop connecting the three branches. Also, the predicted structure does not contain the internal loop in the right branch. Instead, the over-prediction causes two hairpin loops to replace the internal loop from the known structure. The branch on the left hand side is again correctly predicted until the position where the internal loop is supposed to start. After this point, the Nussinov DPA predicts three hairpin loops where the known structure only contains one.

A comparison of Figure 7.5 and 8.2 demonstrates clearly that RnaPredict is able to predict the known structure more accurately than the Nussinov DPA. Qualitatively, RnaPredict's structure resembles the known structure much more than the Nussinov's DPA structure.

8.6 Over-prediction of base pairs

Table 8.6 shows the number of false positive base pairs predicted by RnaPredict and the Nussinov DPA. The data in the table was drawn from the single best Nussinov prediction and the best experiment from RnaPredict for each sequence. The first column gives the name of the sequence. The second shows the weights used for the DPA. The third column shows the number of false positive base pairs predicted by the Nussinov DPA and the fourth column shows the number of false positive base pairs predicted by RnaPredict. The next two columns show the number of correctly predicted base pairs for both Nussinov and RnaPredict. The last column shows which crossover operator/selection strategy/thermodynamic model combination was used to find the structure that predicted the largest number of known base pairs on average.

For all sequences, RnaPredict correctly predicted more base pairs while predicting less false positives. These results indicate that RnaPredict is better at predicting secondary structures of RNA than the Nussinov DPA. Results for additional sequences can be found in Table A.38, on page 153.

Table 8.7 shows a comparison between the single structure with the lowest free energy found with RnaPredict and the Nussinov prediction with the highest number of known base pairs. Again, for all runs, RnaPredict's structures contain more known base pairs and less false predictions than those predicted with Nussinov. Results for additional sequences can be found in Table A.39, on page 154.

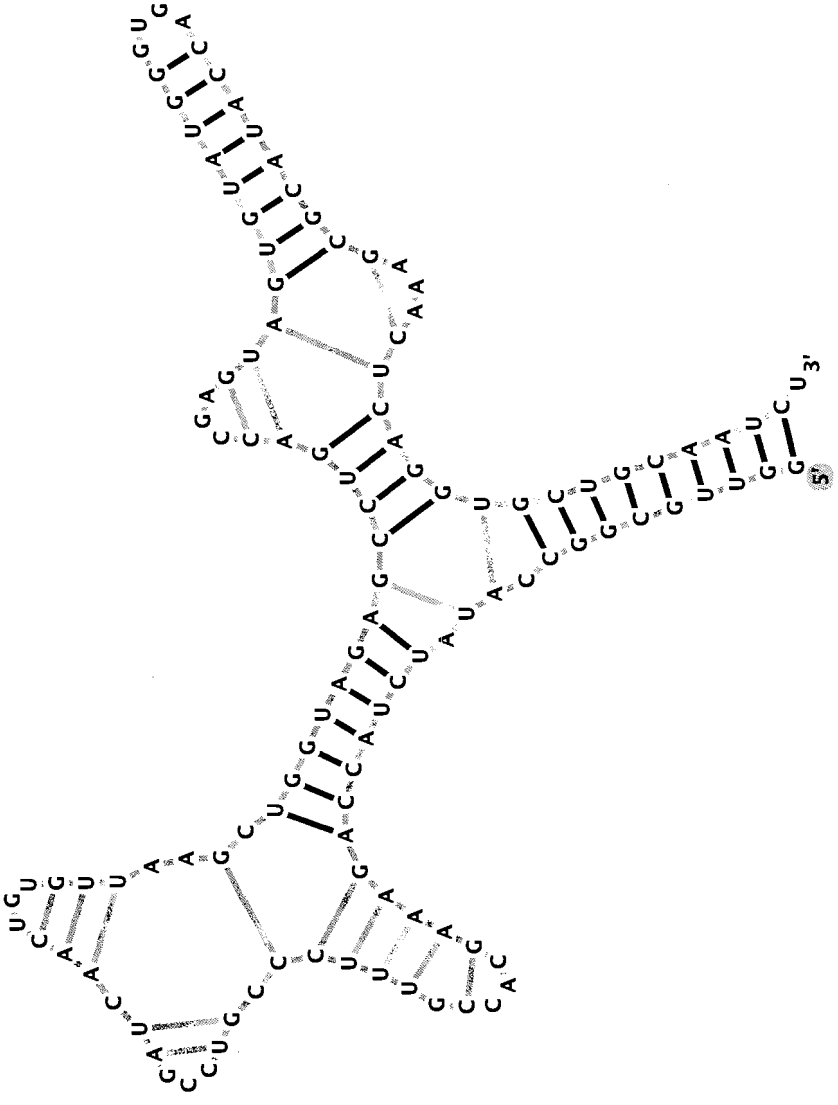

Figure 8.2: The above shows the comparison of the structure predicted with maximal number of base pairs using the Nussinov DPA and the known structure. The light grey base pairs correspond to the predicted structure, while the black ones correspond to the correctly predicted base pairs. The known base pairs were omitted to make the comparison easier. In this case, the Nussinov algorithm was able to predict 75.7% of the known base pairs.

Table 8.6: Comparison between the number of false predictions between best results with the Nussinov DPA and the best experiment with RnaPredict

Sequence	**GC:AU:GU Weights**	**DPA over-pred.**	**GA over-pred.**	**DPA Corr. BPs**	**GA Corr. BPs**	**GA Cross.-Sel.-Model**
X. laevis	3:2:1	286	177.4	47	62.9	CX-STDS-INN
D. virilis	1:1:1	291	195.6	29	43.7	CX-STDS-INN
H. rubra	3:2:1	174	111.6	31	48.4	OX2-STDS-INNHB
H. marismortui	1:1:1	37	14.0	8	16.0	OX2-STDS-INN
S. cerevisiae	1:1:1	17	6.0	28	33.0	CX-STDS-INNHB

Table 8.7: Comparison between the number of false predictions between best results with the Nussinov DPA and the single lowest energy runs with RnaPredict

Sequence	GC:AU:GU Weights	DPA over-pred.	GA over-pred.	DPA Corr. BPs	GA Corr. BPs	GA Cross.-Sel.-Model
X. laevis	3:2:1	286	178	47	77	OX2-STDS-INNHB
D. virilis	1:1:1	291	206	29	39	OX2-STDS-INN
H. rubra	3:2:1	174	102	31	62	OX2-STDS-INNHB
H. marismortui	1:1:1	37	14	8	16	OX2-STDS-INN
S. cerevisiae	1:1:1	17	6	28	33	OX2-STDS-INNHB

Table 8.8 compares the structure with the highest number of correct base pairs predicted by the Nussinov DPA with the single best structure in terms correct base pairs predicted by RnaPredict. The results show that RnaPredict outperforms Nussinov by a large margin for each and every structure. With *Hildenbrandia rubra*, RnaPredict predicts twice as many correct base pairs than Nussinov while predicting less than half of the false positive base pairs. With *Haloarcula marismortui*, the difference is even larger. RnaPredict correctly predicts a structure with three times as many known base pairs and more than 12 times less false positive base pairs. Results for additional sequences can be found in Table A.40, on page 155.

Table 8.8: Comparison between the number of false predictions between best results with the Nussinov DPA and the runs predicting the highest number of known base pairs with RnaPredict

Sequence	**GC:AU:GU Weights**	**DPA over-pred.**	**GA over-pred.**	**DPA Corr. BPs**	**GA Corr. BPs**	**GA Cross.-Sel.-Model**
X. laevis	3:2:1	286	147	47	93	CX-STDS-INN
D. virilis	1:1:1	291	177	29	65	OX2-STDS-INNHB
H. rubra	3:2:1	174	82	31	79	OX2-STDS-INN
H. marismortui	1:1:1	37	3	8	27	PMX-KBR-INNHB
S. cerevisiae	1:1:1	17	6	28	33	OX2-STDS-INNHB

8.7 Chapter summary

The results show that in all cases RnaPredict outperformed the DPA by predicting more base pairs correctly. Furthermore, RnaPredict consistently predicted far less false positive base pairs than the DPA. It is worth noting that for *Haloarcula marismortui* and *Saccharomyces cerevisiae*, RnaPredict results were found repeatedly by more than one parameter setting for the chosen thermodynamic model.

Similarly to RnaPredict, the Nussinov DPA fares better with short sequences. Like RnaPredict, the Nussinov DPA does not model non-canonical base pairs. Larger structures tend to have more non-canonical base pairs making accurate prediction difficult. Also, DPAs cannot model pseudoknots easily. Again, larger structures usually contain a few pseudoknots.

Another reason why RnaPredict found more correct base pairs than the Nussinov DPA is that Nussinov uses base pair maximization. Although it is true that real structures contain a large number of base pairs, they do not contain the maximum number of base pairs. Both the type of base pair and its local environment affect its formation.

Data for additional sequences can be found in Appendix A.

Chapter 9

Comparison to the *mfold* DPA

mfold [5, 6, 7, 8, 9, 10] is a DPA applied to the prediction of secondary structures of RNA. The software uses a complex thermodynamic model for free energy evaluation of structures. The DPA, along with this model, attempts to find the minimum energy structure. Advances in the algorithm also allow to find sub-optimal structures which are important since the natural fold does not always correspond to the global minimum [50, 51, 52].

The current most complete nearest-neighbor thermodynamic model is in *mfold*. The *mfold* package consists of a group of programs that are written in Fortran, C, and C++ tied together with BASH and Perl scripts. The original version was designed to run in the Unix environment. The software was also ported to C++ targeted for the Microsoft Windows platform under the name RNAStructure [11]. The latter implementation offers a point-and-click interface.

mfold uses standard INN-HB parameters [62], but adds modeling for common RNA substructures. These include stacking energies, terminal mismatch stacking energies (hairpin loops), terminal mismatch stacking energies (interior loops), single mismatch energies, 1×2 interior loop energies, tandem mismatch energies, single base stacking energies, loop destabilizing energies, tetra-loops, tri-loops, and other miscellaneous energies.

After generating structures, *mfold* re-evaluates structures with a more complete thermodynamic model using its *efn2* helper application. This software adds a more accurate model of multi-branch loops among other improvements.

To generate the *mfold* results presented here, the *mfold* web server version 3.1 was used. Default settings were used. One noteworthy setting is the percentage of sub-optimality. This percentage allows to control the number of structures predicted by *mfold*. In this

experiment, the value was set to return the 5% lowest energy structures. This corresponds to approximately 20 structures for a 1000 nt sequence.

9.1 *Xenopus laevis* - 945 nt

The longest sequence presented here is *Xenopus laevis*. The results are presented in Table 9.1. Each row in the table represents a single predicted structure. The first column is the free energy optimized by *mfold*. The second column gives the free energy of the predicted structure after being re-evaluated by *efn2*. The third column shows the number of predicted base pairs in the structure. The fourth column shows the number of correctly predicted base pairs, while the last column shows the percentage of known base pairs correctly predicted.

For *Xenopus laevis*, the results show that the lowest energy structure has a free energy of -250.6 kcal/mol. This predicted structure contained 249 base pairs, where 92 were correct. This corresponds to 36.7% of the known structure. The lowest energy structure found with *efn2* had a free energy of -223.49 kcal/mol. Interestingly, the accuracy of this structure was less than that of the structure predicted with the internal, and less accurate, *mfold* thermodynamic model. This structure contained 246 base pairs where 86 were correctly predicted. The percentage of known base pairs correctly predicted was 34.3%.

Lastly, scanning for the structure with highest accuracy, a structure is found with 45.0% of the known base pairs correctly predicted. This structure is ranked fifth in terms of free energy when evaluated with *mfold*'s internal energy model. This structure would normally not be found without the presence of a known structure for comparison.

RnaPredict was not able to perform as well as the *mfold* DPA in the best average case. The best average structure was found with RnaPredict only contained 25.1% of the known base pairs with CX, STDS, and INN. However, the overall lowest energy structure with INN-HB came close to the *mfold* prediction because it contained 32.7% of the known base pairs.

9.2 *Drosophila virilis* - 784 nt

Table 9.2 shows the results obtained for *Drosophila virilis* with *mfold*. The first row of the table shows the minimum free energy structure found. This structure has a free energy of -146.3 kcal/mol and contained 236 base pairs. 15.9% of the known base pairs were correctly

Table 9.1: *Xenopus laevis*, *mfold* results. Number of known base pairs is 251.

mfold ΔG **(kcal/mol)**	***efn2*** ΔG **(kcal/mol)**	**Predicted BP**	**Correctly Predicted BP**	**% Correctly Predicted**
-250.6	-222.85	249	92	36.7
-249.6	-219.75	251	71	28.3
-248.8	-219.63	241	97	38.6
-248.6	-218.69	246	84	33.5
-248.0	-216.51	245	**113**	**45.0**
-248.0	-213.01	242	100	39.8
-247.8	-210.87	241	84	33.5
-247.4	-209.26	243	74	29.5
-247.2	-218.30	246	79	31.5
-247.1	-215.70	244	76	30.3
-246.7	-211.01	238	69	27.5
-246.5	-221.02	244	88	35.1
-246.5	-214.62	245	68	27.1
-246.3	-223.07	248	101	40.2
-245.3	-214.07	250	103	41.0
-245.0	-217.38	248	62	24.7
-244.7	-215.17	243	80	31.9
-244.3	**-223.49**	246	86	34.3
-243.7	-213.42	237	73	29.1
-243.6	-205.90	242	91	36.3
-242.5	-202.27	251	81	32.3

predicted.

The structure with the lowest free energy, with *efn2* was found to have a free energy of −131.55 kcal/mol. This structure contained 254 base pairs and included 14.2% of the base pairs in the natural fold. This new structure actually predicted less correct base pairs than the optimal structure predicted with the internal *mfold* thermodynamic model.

Scanning the table for the highest number of correctly predicted base pairs yields a tie of two structures ranked 19th and 22nd in free energy. These structures greatly improve the last results with 35.2% of the known base pairs correctly predicted. However, in a real experiment, these structures could not have been found because of the lack of a known structure as a basis for comparison.

Comparing with the results from Table 7.5 shows that RnaPredict was able to predict 16.5% of the known base pairs on average using OX2, STDS, and INN. This was the lowest average free energy structure found with INN. Better yet, the experiment with the highest number of correctly predicted base pairs on average was found using CX, STDS and INN. This experiment managed to find 18.8% of the known base pairs on average, but was 1.85 kcal/mol higher in energy than the lowest energy structure, on average.

The single runs from Table 7.6 show that the single lowest energy structures were found with OX2, STDS, and INN. This structure correctly predicted 16.7% of the known base pairs. Similar to the average result, a structure found using with CX, STDS and INN with 2.1 kcal/mol higher in energy contained 24.9% of the known base pairs.

The lowest free energy structure found with *mfold*'s internal thermodynamic model contained fewer correct base pairs at 15.9% overlap. The *efn2* model found even fewer base pairs at 14.2% overlap.

It is interesting to note that even with *mfold*, the results for this sequence were not as good as those for the longer *Xenopus laevis* sequence. Again, this gives evidence that *Drosophila virilis* may be a difficult sequence for the thermodynamic models in both RnaPredict and *mfold*.

9.3 *Hildenbrandia rubra* - 543 nt

Table 9.3 shows the *mfold* results for *Hildenbrandia rubra*. The optimal energy structure found with *mfold* had a free energy of −204.9 kcal/mol. This structure contained 176 base pairs including 35.5% of the base pairs found in the known structure. The structure with

Table 9.2: *Drosophila virilis*, *mfold* results. Number of known base pairs is 233.

mfold ΔG **(kcal/mol)**	***efn2*** ΔG **(kcal/mol)**	**Predicted BP**	**Correctly Predicted BP**	**% Correctly Predicted**
-146.3	-124.43	236	37	15.9
-146.3	-128.56	238	37	15.9
-146.2	-124.07	246	37	15.9
-146.1	-126.92	243	21	9.0
-145.8	-126.59	257	37	15.9
-145.5	-123.19	253	68	29.2
-145.4	-123.30	261	44	18.9
-145.1	-126.92	232	27	11.6
-145.0	-123.57	256	37	15.9
-144.7	-128.43	265	49	21.0
-144.4	-125.39	271	31	13.3
-144.3	-125.03	246	38	16.3
-144.2	-124.69	228	33	14.2
-144.2	-124.04	247	37	15.9
-143.9	-121.32	249	27	11.6
-143.7	-129.30	251	28	12.0
-143.5	-122.97	245	37	15.9
-142.9	-120.17	253	68	29.2
-142.8	-120.26	252	**82**	**35.2**
-142.5	-122.76	230	26	11.2
-142.4	-116.91	237	22	9.4
-142.4	-121.04	255	**82**	**35.2**
-142.3	-123.88	253	38	16.3
-142.1	-126.36	249	21	9.0
-141.8	-118.65	246	79	33.9
-141.4	-125.98	244	28	12.0
-141.2	**-131.55**	254	33	14.2
-141.1	-120.77	242	39	16.7
-140.1	-116.53	243	38	16.3
-140.0	-115.66	235	36	15.4
-139.9	-119.18	249	76	32.6
-139.7	-126.74	260	44	18.9
-139.3	-121.60	246	52	22.3
-139.0	-122.10	238	22	9.4

the lowest free energy according to the *efn2* had a free energy of −199.63 kcal/mol. This structure was less accurate than the former with 171 base pairs predicted and 27.5% of the known ones correctly predicted.

mfold predicted another structure with more correct base pair but with a higher free energy than those listed above. This structure ranked 25th in free energy and contained 167 base pairs. This structure greatly improved on the accuracy from the low energy structure with 60.1% of the known base pairs correctly predicted.

Table 7.8 shows that the structure with average lowest energy found with the INN-HB model has similar accuracy than the lowest energy structure found with *mfold*. The average structure was found with OX2, STDS and INN-HB. It contained 160.1 base pairs where 35.1% of the known base pairs were correctly predicted. Looking at the lowest energy structure found by a single random seed shows that RnaPredict was able to predict even more base pairs correctly. A single run using the same parameters predicted a structure with 164 base pairs with 44.9% of these found in the known structure. In this case, RnaPredict was able to predict more correct base pairs than *mfold* when looking at low energy structures.

9.4 *Haloarcula marismortui* - 122 nt

Table 9.4 shows the *mfold* results for *Haloarcula marismortui*. This sequence is quite short and thus has a much smaller search space and very few structures possible within 5% of the lowest energy structure. In this case, only one structure was found with *mfold*. The structure found contained 34 base pairs where 76.3% overlapped with the known structure.

Table 7.11 shows that, on average, RnaPredict was not able to improve on this result. With OX2, PMX with STDS and INN-HB, RnaPredict was able to predict a structure with 42.1% of the known base pairs with all 30 random seeds. With INN, RnaPredict was able to predict a different structure with similar accuracy with CX, OX2, and PMX using STDS, again, with all 30 random seeds. It is not clear why RnaPredict's average performance was so low in comparison with *mfold*, however RnaPredict was able to find a best structure with 71.1% base pair overlap getting closer to the *mfold* result.

Table 9.3: *Hildenbrandia rubra*, *mfold* results. Number of known base pairs is 138.

mfold ΔG **(kcal/mol)**	***efn2*** ΔG **(kcal/mol)**	**Predicted BP**	**Correctly Predicted BP**	**% Correctly Predicted**
-204.9	-199.11	176	49	35.5
-204.6	**-199.63**	171	38	27.5
-203.9	-191.61	169	53	38.4
-203.4	-191.34	168	61	44.2
-203.3	-195.23	172	71	51.4
-202.6	-198.12	175	40	29.0
-202.3	-184.69	160	47	34.1
-202.0	-191.25	167	42	30.4
-201.7	-191.10	164	65	47.1
-201.5	-183.70	161	72	52.1
-201.5	-195.42	170	57	41.3
-201.1	-191.43	162	46	33.3
-201.0	-186.13	164	68	49.2
-200.8	-188.36	172	57	41.3
-200.8	-185.21	165	50	36.2
-200.6	-183.94	173	67	48.6
-200.3	-193.28	169	55	39.9
-200.2	-194.41	171	64	46.4
-199.9	-192.22	170	42	30.4
-199.9	-190.14	167	57	41.3
-198.7	-190.11	163	40	29.0
-198.5	-191.09	175	66	47.8
-197.7	-186.14	166	44	31.9
-197.0	-188.83	161	65	47.1
-196.6	-183.74	167	**83**	**60.1**
-195.9	-185.17	179	57	41.3
-195.9	-183.87	176	37	26.8
-195.8	-184.93	160	41	29.7
-195.2	-187.12	175	50	36.2

Table 9.4: *Haloarcula marismortui*, *mfold* results. Number of known base pairs is 38.

mfold ΔG **(kcal/mol)**	***efn2*** ΔG **(kcal/mol)**	**Predicted BP**	**Correctly Predicted BP**	**% Correctly Predicted**
-59.5	-56.44	34	29	76.3

9.5 *Saccharomyces cerevisiae* - 118 nt

The *mfold* results for the shortest sequence, *Saccharomyces cerevisiae*, are listed in Table 9.5. Again, due to the smaller search space, only two structures were found. The first row shows the optimal structure found with *mfold* contained as many as 89.2% of the known base pairs in its 41 base pair structure. The lowest energy structure evaluated with *efn2* contained 42 base pairs and 75.7% of the base pairs in the natural fold.

Table 9.5: *Saccharomyces cerevisiae*, *mfold* results. Number of known base pairs is 37.

mfold ΔG **(kcal/mol)**	***efn2*** ΔG **(kcal/mol)**	**Predicted BP**	**Correctly Predicted BP**	**% Correctly Predicted**
-53.5	-50.70	41	**33**	**89.2**
-53.0	**-50.76**	42	28	75.7

On average, RnaPredict was able to match the prediction of *mfold* with all random seeds for each crossover operator and selection strategy with INN-HB. Again, this structure contained 39 base pairs which included 89.2% of the base pairs in the known structure.

9.5.1 Graphical comparison

Both *mfold* and RnaPredict were able to predict 89.2% of the base pairs correctly, but with different structures. Figure 9.1 shows the overlap of these two structures.

The figure shows only the overlap between the two structures (black hydrogen bonds) and the base pairs predicted by *mfold* not present in RnaPredict's prediction (grey hydrogen bonds). The figure shows that both RnaPredict and *mfold* predicted a structure with 89.2% of the known base pairs. However, *mfold* predicted two more base pairs that were not part of the natural fold.

9.6 Over-prediction of base pairs

Table 9.6 shows the number of false-positive base pairs predicted by RnaPredict and the *mfold* DPA. The data in the table was drawn from the lowest energy structure prediction with *mfold*'s internal thermodynamic model and the overall lowest energy structure from RnaPredict for each sequence. The first column gives the sequence. The second and third

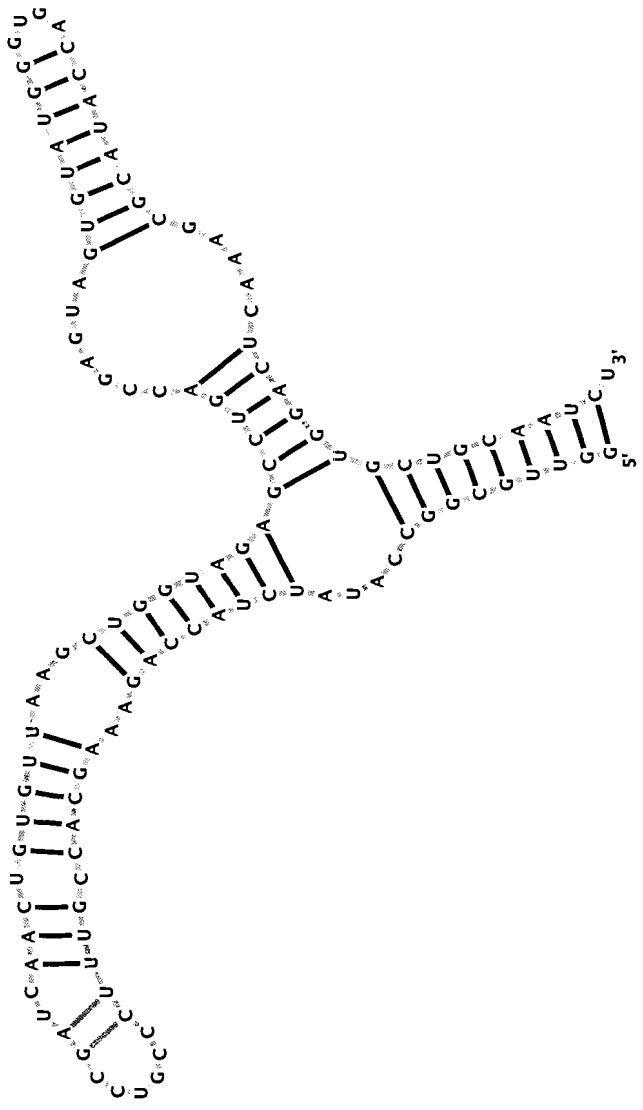

Figure 9.1: The above shows the comparison of the most accurate structure predicted with RnaPredict and the most accurate *mfold* predicted structure. The black base pairs show the overlap between the two structures and the grey base pairs correspond to the extra base pairs predicted by *mfold*. The known base pairs were not added to make the comparison easier. Both structures contained 89.2% of the known base pairs but the *mfold* structure adds two extra false-positive predictions.

columns show the number of false-positive base pairs predicted with the *mfold* DPA and RnaPredict, respectively. The fourth and fifth columns show the number of known base pairs correctly predicted with this structure by *mfold* and RnaPredict. The last column shows which crossover operator/selection strategy/thermodynamic model combination was used to find the structure with the lowest overall energy. Data for additional sequences can be found in Table A.41, on page 156.

Table 9.6: Comparison between the number of false predictions between lowest energy structure found with the *mfold* DPA and the overall lowest energy single RnaPredict runs

Sequence	**DPA over-pred.**	**GA over-pred.**	**DPA Corr. BPs**	**GA Corr. BPs**	**GA Cross.-Sel.-Model**
X. laevis	157	178	92	77	OX2-STDS-INNHB
D. virilis	199	206	37	39	OX2-STDS-INN
H. rubra	127	102	49	62	OX2-STDS-INNHB
H. marismortui	5	14	29	16	OX2-STDS-INN
S. cerevisiae	8	6	33	33	OX2-STDS-INNHB

The results show that RnaPredict outperformed the DPA by predicting more base pairs correctly with two sequences, *Drosophila virilis* and *Hildenbrandia rubra*. RnaPredict also performed as well as the DPA with the shortest sequence, *Saccharomyces cerevisiae*. Furthermore, RnaPredict predicted less false-positive base pairs with two sequences, *Hildenbrandia rubra* and *Saccharomyces cerevisiae*.

The final comparison will be between the overall highest number of correct base pairs predicted by both *mfold* and RnaPredict, regardless of energy. Table 9.7 shows the same format as Table 9.6. The results show that although *mfold* is able to predict more base pairs correctly in four sequences (*Xenopus laevis*, *Drosophila virilis*, *Hildenbrandia rubra*, and *Haloarcula marismortui*), RnaPredict predicts less false-positives in three sequences (*Hildenbrandia rubra*, *Haloarcula marismortui*, and *Saccharomyces cerevisiae*). Data for other sequences can be found in Table A.42, on page 157.

Table 9.7: Comparison between the number of false predictions between best structure with the *mfold* DPA and the overall best single structure found with RnaPredict

Sequence	DPA over-pred.	GA over-pred.	DPA Corr. BPs	GA Corr. BPs	GA Cross.-Sel.-Model
X. laevis	132	147	113	93	CX-STDS-INN
D. virilis	170	177	82	65	OX2-STDS-INNHB
H. rubra	84	82	83	79	OX2-STDS-INN
H. marismortui	5	3	29	27	PMX-KBR-INNHB
S. cerevisiae	8	6	33	33	OX2-STDS-INNHB

9.7 Chapter summary

The *mfold* DPA performs better than the Nussinov DPA by using a complex thermodynamic model and optimizing free energy instead of maximizing the number of possible base pairs. This method predicted far more correct base pairs for each sequence studied.

Predicted structures for short sequences had a better overlap than those from longer sequences. Again, non-canonical base pairs, pseudoknots and long range intermolecular interactions can not be modeled with the current *mfold* DPA.

RnaPredict performed very well predicting more correct base pairs with the *Drosophila virilis* and *Hildenbrandia rubra* sequences than *mfold* when comparing lowest energy structures (Table 9.6) while keeping false predictions low. RnaPredict also predicted as many correct base pairs with *Saccharomyces cerevisiae.*

In the best case, RnaPredict predicted as many correct base pairs as *mfold* with the *Saccharomyces cerevisiae* sequence while predicting less false-positive base pairs. RnaPredict also predicted less false-positive base pairs than *mfold* for *Hildenbrandia rubra*, *Haloarcula marismortui*, and *Saccharomyces cerevisiae.* Data for additional sequences can be found in Appendix A.

These results are quite encouraging showing that a GA with a simplistic thermodynamic model performs as well as a mature DPA with a complex thermodynamic model.

Chapter 10

Conclusion

This document discusses the research done developing a GA for RNA secondary structure prediction through energy minimization. The current version is a complete redesign and reimplementation, in object-oriented C++, inspired from a previously version written in Dr. Wiese's lab. The focus of the work presented here was the implementation of three new thermodynamic models. One of these models, Mathews, was a hydrogen bond model which associates free energy changes to single base pairs. The other two models were stacking energy models, INN and INN-HB, attributing free energy changes to tandem base pairs.

Several objectives were met during the course of this research. Optimized GA settings for RnaPredict were determined to improve prediction of low energy structure. Different crossover and mutation rate combinations were systematically examined to determine optimal settings. Selection techniques, STDS and KBR, were also tested with various crossover rates. Encoding, binary and permutation, was studied along with corresponding crossover operators. The quality of the predicted structures was compared to known structures and to those generated by the Nussinov DPA and the *mfold* DPA.

A typical run on a Pentium 4 2.6 GHz computer with 1.5 GB of RAM running Linux 2.4 for a 1000 nucleotide sequence can run for 10 hours with the slowest operator, ASERC, to 15 minutes for the fastest operator, 1-Point. For such a sequence, the RnaPredict application can consume as much as 500 Mb of memory.

Eleven sequences were tested: *Sulfolobus acidocaldarius* (1494 nt), *Homo sapiens* (954 nt), *Xenopus laevis* (945 nt), *Drosophila virilis* (784 nt), *Caenorhabditis elegans* (697 nt), *Acanthamoeba griffini* (556 nt), *Hildenbrandia rubra* (543 nt), *Aureoumbra lagunensis* (468 nt), *Haloarcula marismortui* (122 nt), *Arthrobacter globiformis* (123 nt), and *Saccharomyces*

cerevisiae (118 nt). Of these sequences, five were discussed in detail: *Xenopus laevis*, *Drosophila virilis*, *Hildenbrandia rubra*, *Haloarcula marismortui*, and *Saccharomyces cerevisiae*. Data for the remaining sequences can be found in Appendix A.

The results show that while sampling structures in the search space, a strong correlation was found between the structures' free energy and the number of correct base pairs predicted with stacking energy models such as INN and INN-HB. For the sequences discussed in detail in this document, the correlation coefficient was very close to -1, especially with INN-HB. These results showed clearly that the stacking energy models outperform the hydrogen bond models.

Next, different encodings, selection strategies, crossover operators, crossover rates, and mutation rates were tested. With both selection strategies, permutation encoding found lower energy structures more often than binary encoding. OX2 and CX were found to yield lower energy structures than with any other crossover operator when coupled with STDS, high crossover rates, and high mutation rates. With PMX, lower energy structures were found with KBR with high crossover rates and high mutation rates as compared to those found with PMX and STDS using the same crossover and mutation rates. However, these PMX experiments did not perform as well as the best OX2 and CX runs. To make the results uniform, a crossover rate of 0.7 was coupled with a mutation rate of 0.8 for all runs in subsequent experiments.

Once the thermodynamic models were partially validated through the computation of the correlation coefficient and the behavior of the GA parameter settings were controlled, experiments were done on real structures in the hope to find the natural fold.

RnaPredict was able to partially predict the structure of large sequences. For instance, with *Xenopus laevis*, the highest number of base pairs found by a single run was 37.1% with CX, STDS, and INN-HB. These results were greatly improved with *Hildenbrandia rubra*, where 57.2% of the base pairs were correctly predicted by a single run using OX2, STDS, and INN. For the shortest sequence, *Saccharomyces cerevisiae*, RnaPredict was able to predict as many as 89.2% of the correct base pairs. This turned out to be the theoretical maximum that the helix generation model allows. Effectively, it can be said that RnaPredict found 100% of the correct structure within the constraints of its internal model and therefore represents a very effective search engine.

When comparing with the Nussinov DPA, RnaPredict found more correct base pairs for every sequence tested. Furthermore, RnaPredict predicts far fewer false positive base pairs

for every sequence. These results show that RnaPredict efficiently searches the conformational space and does not simply predict structures that only have a large amount of base pairs.

When comparing with the *mfold* DPA, RnaPredict performed competitively. A direct comparison of the lowest free energy structure found with *mfold* and the lowest free energy structure found with RnaPredict shows that *mfold* was able to predict more base pairs correctly for two sequence, *Xenopus laevis* and *Haloarcula marismortui*, while RnaPredict predicted more correct base pairs with two sequences, *Drosophila virilis* and *Hildenbrandia rubra*, and predicted equally as many base pairs with *Saccharomyces cerevisiae*. RnaPredict predicted less false positive base pairs with *Saccharomyces cerevisiae* and *Hildenbrandia rubra* than *mfold*.

Lastly, the structures with the highest number of base pairs correctly predicted with both *mfold* and RnaPredict, regardless of energy, were compared. The results show that both RnaPredict and *mfold* were able to predict equally as many base pairs correctly with *Saccharomyces cerevisiae*. Also, RnaPredict predicted less false positive base pairs than *mfold* with the following sequences, *Hildenbrandia rubra*, *Haloarcula marismortui*, and *Saccharomyces cerevisiae*.

In modelling RNA secondary structure with RnaPredict, the following assumptions are made. It uses a simplistic, but strict, helix generation model where helices can contain no less than three adjacent base pairs and must be connected by three or more nucleotides. Also, the free energy of the structures is computed using a simplistic thermodynamic model that approximates the free energy by only modelling adjacent base pairs. The decoder disabled the formation of pseudoknots to create simpler structures. With all these constraints and limitations, the results are still very significant and demonstrate the usefulness of EAs in the field of secondary structure prediction of RNA.

10.1 Future work

This document describes the advances made during the course of this research. Various other improvements are possible and could dramatically improve the results of the GA.

10.1.1 Non-canonical base pairs

The inclusion of non-canonical base pairs would allow RnaPredict to predict structures more accurately. Most structures contain some non-canonical base pairs. The problem with predicting these base pairs is that it is difficult to determine which base pairs will form. It is possible that a base pair of a particular identity form in one structure but not in another. Preliminary research can be found in [119, 120, 121]

10.1.2 Modelling common RNA substructures

Another goal is to explicitly model common RNA substructures. The current thermodynamic models model stacks only. Stacks occur when a set of adjacent base pairs form. The presence of these stacks forms higher order substructures such as loops and bulges. Different substructures have different energy contributions to a structure. Free energy parameters have been devised to model these contributions. A review [53] provides a good starting point to incorporate substructure modelling into RnaPredict.

With a complete thermodynamic model such as the one in *efn2*, it would be possible to compare the results, obtained by RnaPredict, fairly and directly with those from *mfold*.

10.1.3 Optimizing code

The current implementation of RnaPredict has some performance issues. Some code sections have been identified as bottlenecks making the algorithm run slowly and consume large amounts of memory. Most of the code has been written following specifications directly from the literature with little or no optimization.

Algorithm performance could be easily improved by formally profiling the code, and optimizing the problem code sections. A good place to start would be optimizing the crossover operators as these are executed repeatedly during a run consuming a large part of the algorithm's runtime.

10.1.4 Fitness scaling

RnaPredict uses STDS, or roulette-wheel selection [103], to select individuals from the population to undergo crossover and mutation. The implementation of STDS in RnaPredict is done by giving a pie-shaped slice of a roulette-wheel proportional to its absolute fitness. The wheel is spun and an individual is chosen if the wheel stops on its slice.

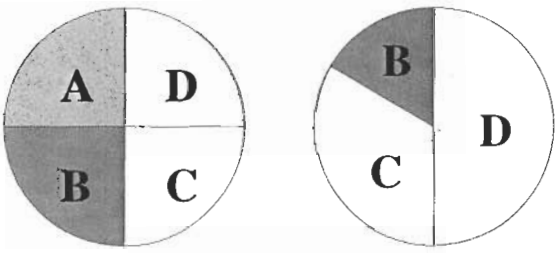

Figure 10.1: The roulette-wheel on the left shows how pie-shaped slices are assigned for a population of four individuals where $f(A) = 1000$, $f(B) = 1001$, $f(C) = 1002$, and $f(D) = 1003$ when using absolute fitness scaling. Each slice is approximately the same size and each individual has approximately the same chance of being chosen during selection. On the right, relative fitness scaling is used (i.e.: subtract fitness of the least fit individual from each individual's fitness). Proportional pie-shaped slices are given to each individuals relative fitness resulting in $f(A) = 0$, $f(B) = 1$, $f(C) = 2$, and $f(D) = 3$. In this case, the individual with highest fitness has a larger slice and therefore a larger probability of being chosen.

Which individual is chosen with STDS is highly dependent on fitness scaling. With linear fitness scaling, a slice on the roulette-wheel is proportional to the individual's fitness minus the fitness of the least fit individual, relative to the sum of all other fitnesses computed in the same fashion. Currently, when the population converges to highly fit individuals, RnaPredict's implementation selects randomly from the population. By properly implementing fitness scaling using relative fitness, highly fit individuals have a greater chance of being selected when the population converges, as was intended by the author [103].

10.1.5 Selection

Currently, RnaPredict has the option of using STDS or KBR. A new feature that should be added to RnaPredict is the option of tournament selection. Tournament selection [122] is done by choosing a number of individuals randomly from the population and the best individual from this group is chosen as the parent. The process is repeated for a second parent. Tournament selection has one parameter which is the tournament size. The tournament size can take any value ranging from 2 to the population size.

10.1.6 Modelling pseudoknots

A GA is able to predict pseudoknots, but for the experiments presented here, the formation of pseudoknots was disabled in the decoder. This was done because RnaPredict cannot currently model the free energy contribution of pseudoknots and allowed too many of them to form. Most large structures contain one or more pseudoknots. Allowing pseudoknots to form and modelling them properly would improve the accuracy of prediction of structures. Thermodynamic parameters for pseudoknots exist. They can be found in [90], [91], and [92].

DPAs, like Nussinov and *mfold*, have great difficulty generating structures with pseudoknots due to their inherent design. Modelling pseudoknots with RnaPredict would be quite advantageous. With proper parameters, it should be possible to improve prediction beyond what is possible with *mfold*.

RNAML

A problem that arises often with computer applications is the large number of different formats used to describe data. Even with something as simple as a secondary structure of RNA, there are numerous formats available.

RnaPredict has used the Connectivity Table (CT) file format [10] for output of all structures. CT has been the standard file format used to represent secondary structures of RNA. It has been the preferred format used by *mfold* for output. However, the CT file format has various slightly different implementations making it difficult to write a robust universal parser.

The main source of known structures used in this research came from the CRW website which uses the Base Pair Sequence (BPSeq) format, a modified CT format. This format simply removes redundant columns found in the CT file format.

Another class of formats that exists is the Dot Bracket Notation (DBN) file format. This originated from the simple idea of encoding RNA secondary structures using two strings. The first string is the primary sequence while the second encodes for the secondary structure using '(' to indicate a bound nucleotide on the 5′-end, ')' for a bound nucleotide on the 3′-end and a ':' for an unpaired nucleotide. A modified DBN format [123] uses '[' and ']' to represent pseudoknots.

Most visualization applications are able to read the CT file format. Inclusion of the CT file format to RnaPredict was the most natural design decision. The format was chosen to

make RnaPredict interoperate with the largest number of external applications.

RNAML [124] has been proposed as a standard file format for communicating RNA structural data. RNAML complies with eXtensible Markup Language (XML) which is a widely accepted syntax standard. The format can be used to represent structural information for the primary, secondary, and tertiary structure of RNA. It can describe base pairs, base triples, and pseudoknots. Because of the nature of XML, RNAML can be extended to describe different types of data.

RNAML has numerous advantages. An RNAML file can be used to model a single structure or any number of structures. Such a feature could be useful to generate more cohesive data where the RNAML could contain all of the data from a particular run.

Implementing a robust parser for RNAML should be a trivial exercise using any language with XML programming toolkits. Outputting RNAML structures is also a simple task. The RNAML format uses a logical structure without the constraints of delimiter-separated value (DSV) formats. Simply ensuring the logical rules are met is enough to generate a valid structure.

10.1.7 Seeding the random population

The random population is generated by RnaPredict in the first generation. In the usual case, the GA finds lower energy structures with each generation. To improve the convergence velocity and generate higher quality structures, the GA's random population could be seeded with lower energy structures.

mfold can be used to generate a large number of low energy in a short amount of time. The structures generated by *mfold* can be used to seed the random population thus potentially improving the results from RnaPredict.

10.1.8 Other improvements

A GA is a stochastic algorithm using a wide variety of parameters to control it. The results of the GA could be improved by testing different population sizes and number of generations. Currently, the population size was set to 700 for all experiments. This was chosen as a reasonable number, but no other sizes have been tested. It is possible that changing the size could influence the results.

The number of generations was set to 700 for most sequences. It was noticed that

different sequence lengths require different numbers of generations for the population to converge. For a short sequence like *Saccharomyces cerevisiae*, less than ten generations are required while experiments using long sequences can keep making progress for thousands of generations.

All experiments reported in this thesis ran with 1-elitism. However, earlier work [34] suggests that under some circumstances KBR without elitism outperformed KBR with 1-elitism. More experiments could be done to determine whether KBR without elitism could improve on the results presented here.

Different crossover operators should be implemented and tested. Many operators exist in the literature, such as OX3 [125], MPX [126], MX1 [127], and MX2 [127]. The improvements could yield better results or higher performance.

Appendix A

Data for other sequences

This appendix contains the data tables for the sequences that were not discussed in detail throughout the main chapters.

A.1 Correlation data

Here is the correlation data for all the sequences studied.

Table A.1: The correlation between the free energy of structures and the number of correctly predicted base pairs.

Sequence	INNHB	INN	MAJOR	MATHEWS
S. cerevisiae	**-0.98**	-0.96	-0.15	-0.78
X. laevis	**-0.96**	-0.90	-0.78	-0.58
H. rubra	**-0.94**	-0.87	0.36	-0.71
S. acidocaldarius	**-0.93**	-0.88	-0.26	-0.76
D. virilis	**-0.93**	-0.50	-0.18	-0.71
H. sapiens	**-0.81**	-0.77	-0.29	0.20
A. lagunensis	-0.76	-0.77	-0.31	**-0.78**
A. globiformis	-0.76	**-0.88**	-0.83	-0.72
H. marismortui	-0.74	**-0.86**	-0.56	-0.30
A. griffini	-0.74	**-0.84**	-0.70	-0.36
C. elegans	-0.26	-0.74	0.08	**-0.80**

A.2 *Sulfolobus acidocaldarius* - 1494 nt

Table A.2: *Sulfolobus acidocaldarius* details

Filename	d.16.a.S.acidocaldarius.bpseq
Organism	*Sulfolobus acidocaldarius*
Accession Number	D14876
Class	16S rRNA
Length	1494 nucleotides
# of BPs in known structure	468
# of non-canonical base pairs	22

Table A.3: Results of comparison with known *Sulfolobus acidocaldarius* structure grouped by thermodynamic model. The known structure contains 468 base pairs. Each row represents an experiment consisting of 30 averaged runs.

ΔG (kcal/mol)	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-658.28	427.0	89.2	19.1	CX	STDS[a]	INNHB
-651.91	422.0	**89.4**	**19.1**	OX2	STDS	INNHB
-612.01	411.7	71.5	15.3	OX2	KBR	INNHB
-601.38	407.5	63.9	13.6	CX	KBR	INNHB
-594.13	405.9	57.9	12.4	PMX	KBR	INNHB
-497.58	376.1	36.9	7.9	PMX	STDS	INNHB
-620.2	427.6	**92.0**	**19.7**	CX	STDS	INN
-607.4	418.9	89.7	19.2	OX2	STDS	INN
-573.9	411.0	66.1	14.1	OX2	KBR	INN
-567.7	411.1	62.4	13.3	PMX	KBR	INN
-560.2	408.7	56.5	12.1	CX	KBR	INN
-478.2	378.9	39.4	8.4	PMX	STDS	INN

[a]All STDS runs were extended to 1400 generations to improve convergence

Table A.4: Best results of comparison with known *Sulfolobus acidocaldarius* structure grouped by thermodynamic model. The known structure contains 468 base pairs. Best single run ranked by free energy.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-691.24	1	1286	439	**131**	**28.0**	CX	STDS[a]	INNHB
-687.51	1	1391	441	115	24.6	OX2	STDS	INNHB
-653.97	1	678	421	85	18.2	OX2	KBR	INNHB
-646.96	1	693	414	114	24.4	CX	KBR	INNHB
-630.49	1	640	426	71	15.2	PMX	KBR	INNHB
-537.83	1	1395	385	45	9.6	PMX	STDS	INNHB
-652.5	1	1254	435	105	22.4	CX	STDS	INN
-652.1	1	1368	441	**138**	**29.5**	OX2	STDS	INN
-644.7	1	692	430	93	19.9	PMX	KBR	INN
-612.7	1	697	416	89	19.0	OX2	KBR	INN
-610.0	1	674	427	72	15.4	CX	KBR	INN
-523.7	1	1380	398	51	10.9	PMX	STDS	INN

[a]All STDS runs were extended to 1400 generations to improve convergence

Table A.5: Single run with highest number of correctly predicted base pairs of *Sulfolobus acidocaldarius*, regardless of free energy grouped by thermodynamic model. The known structure contains 468 base pairs.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-691.24	1	1286	439	**131**	**28.0**	CX	STDS[a]	INNHB
-682.97	1	1367	439	**131**	**28.0**	OX2	STDS	INNHB
-646.96	1	693	414	114	24.4	CX	KBR	INNHB
-635.38	1	698	419	99	21.2	OX2	KBR	INNHB
-615.72	1	626	408	88	18.8	PMX	KBR	INNHB
-523.51	1	1393	399	66	14.1	PMX	STDS	INNHB
-652.1	1	1368	441	138	29.5	OX2	STDS	INN
-633.4	1	1385	432	**144**	**30.8**	CX	STDS	INN
-595.7	1	583	413	107	22.9	OX2	KBR	INN
-584.5	1	689	416	110	23.5	PMX	KBR	INN
-516.5	1	674	427	80	17.1	CX	KBR	INN
-516.5	1	1237	393	69	14.7	PMX	STDS	INN

[a]All STDS runs were extended to 1400 generations to improve convergence

Table A.6: *Sulfolobus acidocaldarius*, Nussinov results. Number of known base pairs is 468.

GC:AU:GU Weights	**Predicted BP**	**Correctly Predicted BP**	**Correctly Predicted (%)**
1:1:1	584	187	39.9
3:2:1	570	143	30.5
3:2:2	582	187	39.9

Table A.7: *Sulfolobus acidocaldarius*, *mfold* results. Number of known base pairs is 468.

mfold **ΔG (kcal / mol)**	***efn2*** **ΔG (kcal / mol)**	**Predicted BP**	**Correctly Predicted BP**	**% Correctly Predicted**
-822.9	-781.20	494	261	55.8
-821.8	-773.67	493	243	51.9
-821.3	**-787.66**	496	266	56.8
-820.6	-777.39	496	**271**	**57.9**
-817.5	-766.00	493	240	51.3
-816.7	-779.52	487	270	57.7
-816.6	-766.23	495	285	60.9
-816.1	-774.22	485	247	52.8
-815.7	-779.32	494	243	51.9
-815.6	-779.82	492	237	50.6
-815.2	-776.33	489	249	53.2
-814.8	-761.41	491	230	49.1
-814.5	-768.46	495	243	51.9
-813.9	-762.55	491	229	48.9
-813.5	-772.38	490	254	54.3
-813.0	-783.78	489	241	51.5

A.3 *Homo sapiens* - 954 nt

Table A.8: *Homo sapiens* details

Filename	d.16.m.H.sapiens.bpseq
Organism	*Homo sapiens*
Accession Number	J01415
Class	16S rRNA
Length	954 nucleotides
# of BPs in known structure	266
# of non-canonical base pairs	30

Table A.9: Results of comparison with known *Homo sapiens* structure grouped by thermodynamic model. The known structure contains 266 base pairs. Each row represents an experiment consisting of 30 averaged runs.

ΔG (kcal / mol)	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-273.10	238.7	47.9	18.0	CX	STDS[a]	INNHB
-272.06	238.7	**48.1**	**18.1**	OX2	STDS	INNHB
-256.66	232.4	37.0	13.9	CX	KBR	INNHB
-254.40	232.4	36.3	13.6	OX2	KBR	INNHB
-253.69	232.2	35.3	13.3	PMX	KBR	INNHB
-222.60	223.3	28.6	10.7	PMX	STDS	INNHB
-267.4	243.3	45.7	17.2	OX2	STDS	INN
-260.3	239.9	**46.5**	**17.5**	CX	STDS	INN
-250.7	238.0	35.6	13.4	OX2	KBR	INN
-248.2	236.2	33.4	12.6	CX	KBR	INN
-245.7	234.4	35.3	13.3	PMX	KBR	INN
-215.9	226.6	26.1	9.8	PMX	STDS	INN

[a]All STDS runs were extended to 1000 generations to improve convergence

Table A.10: Best results of comparison with known *Homo sapiens* structure grouped by thermodynamic model. The known structure contains 266 base pairs. Best single run ranked by free energy.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-289.10	1	981	245	72	27.1	OX2	STDS[a]	INNHB
-288.29	1	997	251	**74**	**27.8**	CX	STDS	INNHB
-280.28	1	613	244	24	9.0	PMX	KBR	INNHB
-275.68	1	473	239	50	18.8	OX2	KBR	INNHB
-271.13	1	654	237	64	24.1	CX	KBR	INNHB
-249.70	1	891	236	34	12.8	PMX	STDS	INNHB
-276.2	1	996	253	**65**	**24.43**	OX2	STDS	INN
-275.1	1	696	238	50	18.79	OX2	KBR	INN
-274.7	1	909	244	59	22.18	CX	STDS	INN
-266.7	1	624	244	35	13.15	CX	KBR	INN
-261.1	1	554	241	42	15.78	PMX	KBR	INN
-231.9	1	863	236	19	7.14	PMX	STDS	INN

[a]All STDS runs were extended to 1000 generations to improve convergence

Table A.11: Single run with highest number of correctly predicted base pairs of *Homo sapiens*, regardless of free energy grouped by thermodynamic model. The known structure contains 266 base pairs.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-282.89	1	891	236	78	29.3	CX	STDS[a]	INNHB
-280.42	1	963	250	**89**	**33.5**	OX2	STDS	INNHB
-271.13	1	654	237	64	24.1	CX	KBR	INNHB
-257.36	1	608	233	71	26.7	OX2	KBR	INNHB
-255.31	1	623	231	67	25.2	PMX	KBR	INNHB
-230.14	1	993	241	60	22.6	PMX	STDS	INNHB
-275.4	1	999	248	70	26.3	OX2	STDS	INN
-263.6	1	939	248	**82**	**30.8**	CX	STDS	INN
-254.2	1	234	240	57	21.4	PMX	KBR	INN
-252.6	2	500	238	64	24.1	OX2	KBR	INN
-248.5	1	671	232	59	22.2	CX	KBR	INN
-225.7	1	993	229	50	18.8	PMX	STDS	INN

[a]All STDS runs were extended to 1000 generations to improve convergence

Table A.12: *Homo sapiens*, Nussinov results. Number of known base pairs is 266.

GC:AU:GU Weights	**Predicted BP**	**Correctly Predicted BP**	**Correctly Predicted (%)**
1:1:1	342	33	12.4
3:2:1	333	22	8.2
3:2:2	339	32	12.0

Table A.13: *Homo sapiens*, *mfold* results. Number of known base pairs is 266.

mfold ΔG **(kcal / mol)**	***efn2*** ΔG **(kcal / mol)**	**Predicted BP**	**Correctly Predicted BP**	**% Correctly Predicted**
-250.9	-217.20	258	**95**	**35.7**
-250.5	**-222.51**	255	91	34.2
-248.7	-213.42	251	84	31.6
-247.5	-213.14	262	44	16.5
-247.2	-219.97	251	51	19.2
-247.2	-214.50	256	82	30.8
-247.0	-207.98	260	52	19.5
-246.7	-210.39	255	82	30.8
-246.3	-207.85	257	81	30.5
-246.0	-206.53	256	37	13.9
-244.8	-217.55	255	57	21.4
-243.7	-205.51	262	43	16.2
-243.4	-198.70	259	50	18.8
-243.2	-211.12	258	67	25.2
-243.1	-198.17	258	44	16.5
-241.6	-202.39	245	51	19.2
-241.5	-210.26	257	59	22.2
-240.9	-219.87	259	50	18.8
-240.8	-204.14	266	57	21.4

A.4 *Caenorhabditis elegans* - 697 nt

Table A.14: *Caenorhabditis elegans* details

Filename	d.16.m.C.elegans.bpseq
Organism	*Caenorhabditis elegans*
Accession Number	X54252
Class	16S rRNA
Length	697 nucleotides
# of BPs in known structure	189
# of non-canonical base pairs	23

Table A.15: Results of comparison with known *Caenorhabditis elegans* structure grouped by thermodynamic model. The known structure contains 189 base pairs. Each row represents an experiment consisting of 30 averaged runs.

ΔG (kcal / mol)	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-166.14	207.8	**27.3**	**14.5**	OX2	STDS	INNHB
-163.49	204.1	25.3	13.4	CX	STDS	INNHB
-155.30	200.7	22.8	12.1	OX2	KBR	INNHB
-151.28	202.1	21.2	11.2	CX	KBR	INNHB
-150.84	199.8	22.4	11.8	PMX	KBR	INNHB
-131.28	192.3	16.3	8.6	PMX	STDS	INNHB
-147.3	203.3	**30.9**	**16.4**	OX2	STDS	INN
-147.1	201.1	30.2	16.0	CX	STDS	INN
-134.8	197.2	22.1	11.7	OX2	KBR	INN
-133.7	196.7	21.1	11.2	PMX	KBR	INN
-132.8	194.3	23.4	12.4	CX	KBR	INN
-116.1	190.6	19.6	10.4	PMX	STDS	INN

Table A.16: Best results of comparison with known *Caenorhabditis elegans* structure grouped by thermodynamic model. The known structure contains 189 base pairs. Best single run ranked by free energy.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-179.20	1	673	212	**37**	**19.6**	OX2	STDS	INNHB
-175.23	1	151	206	30	15.9	OX2	KBR	INNHB
-174.11	1	520	223	17	9.0	CX	STDS	INNHB
-173.40	1	662	212	20	10.6	PMX	KBR	INNHB
-163.32	1	537	198	10	5.3	CX	KBR	INNHB
-154.96	1	687	211	27	14.3	PMX	STDS	INNHB
-161.6	1	597	204	34	18.0	OX2	STDS	INN
-157.8	1	447	208	**35**	**18.5**	CX	STDS	INN
-147.3	1	462	199	27	14.2	OX2	KBR	INN
-146.0	1	646	197	**35**	**18.5**	PMX	KBR	INN
-144.3	1	666	200	15	7.9	CX	KBR	INN
-134.3	1	672	199	28	14.8	PMX	STDS	INN

Table A.17: Single run with highest number of correctly predicted base pairs of *Caenorhabditis elegans*, regardless of free energy grouped by thermodynamic model. The known structure contains 189 base pairs.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-164.58	1	609	202	38	20.1	OX2	STDS	INNHB
-162.61	1	649	203	40	21.2	CX	STDS	INNHB
-161.78	1	598	203	**49**	**25.9**	CX	KBR	INNHB
-150.96	1	606	199	43	22.8	PMX	KBR	INNHB
-143.12	1	526	216	41	21.7	OX2	KBR	INNHB
-125.89	1	654	192	42	22.2	PMX	STDS	INNHB
-156.7	1	656	202	**55**	**29.1**	OX2	STDS	INN
-153.7	1	561	209	57	30.2	CX	STDS	INN
-144.3	1	666	200	15	7.9	CX	KBR	INN
-128.8	1	517	201	44	23.3	OX2	KBR	INN
-126.2	1	542	185	39	20.6	PMX	KBR	INN
-125.4	1	695	191	39	20.6	PMX	STDS	INN

Table A.18: *Caenorhabditis elegans*, Nussinov results. Number of known base pairs is 189.

GC:AU:GU Weights	**Predicted BP**	**Correctly Predicted BP**	**Correctly Predicted (%)**
1:1:1	284	10	5.2
3:2:1	275	20	10.5
3:2:2	281	26	13.7

Table A.19: *Caenorhabditis elegans*, *mfold* results. Number of known base pairs is 189.

mfold **ΔG (kcal / mol)**	***efn2*** **ΔG (kcal / mol)**	**Predicted BP**	**Correctly Predicted BP**	**% Correctly Predicted**
-142.1	-125.22	217	**40**	**21.2**
-141.3	-123.20	219	32	16.9
-141.2	-124.04	216	**40**	**21.2**
-140.6	-124.20	211	**40**	**21.2**
-137.9	-126.18	221	25	13.2
-137.6	-121.46	219	25	13.2
-137.5	-123.11	216	**40**	**21.2**
-137.3	-121.59	213	**40**	**21.2**
-137.0	-122.10	211	20	10.6
-136.8	-123.68	212	37	19.6
-136.4	-118.90	211	27	14.3
-136.2	-126.56	221	25	13.2
-136.2	**-128.97**	216	20	10.6
-136.1	-115.94	200	27	14.3
-135.9	-117.46	206	32	16.9
-135.7	-120.06	208	35	18.5
-135.5	-118.87	206	27	14.3
-135.5	-120.81	216	37	19.6
-135.4	-122.56	218	35	18.5
-135.1	-125.99	213	20	10.6

A.5 *Acanthamoeba griffini* - 556 nt

Table A.20: *Acanthamoeba griffini* details

Filename	b.I1.e.A.griffini.1.C1.SSU.516.bpseq
Organism	*Acanthamoeba griffini*
Accession Number	U02540
Class	Group I intron, 16S rRNA
Length	556 nucleotides
# of BPs in known structure	131
# of non-canonical base pairs	1

Table A.21: Results of comparison with known *Acanthamoeba griffini* structure grouped by thermodynamic model. The known structure contains 131 base pairs. Each row represents an experiment consisting of 30 averaged runs.

ΔG (kcal / mol)	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-195.47	160.2	**44.0**	**33.6**	OX2	STDS	INNHB
-192.52	158.3	42.2	32.2	CX	STDS	INNHB
-184.52	156.9	35.5	27.1	PMX	KBR	INNHB
-183.27	156.7	33.5	25.6	CX	KBR	INNHB
-183.12	155.5	34.1	26.1	OX2	KBR	INNHB
-162.23	149.1	25.8	19.7	PMX	STDS	INNHB
-180.8	164.7	**45.2**	**34.5**	OX2	STDS	INN
-179.4	163.3	44.1	33.7	CX	STDS	INN
-170.6	160.8	34.9	26.7	OX2	KBR	INN
-169.5	159.9	34.2	26.1	PMX	KBR	INN
-166.7	157.7	34.3	26.2	CX	KBR	INN
-149.8	150.7	20.3	15.5	PMX	STDS	INN

Table A.22: Best results of comparison with known *Acanthamoeba griffini* structure grouped by thermodynamic model. The known structure contains 131 base pairs. Best single run ranked by free energy.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-205.31	1	422	159	47	35.9	CX	STDS	INNHB
-203.01	1	616	158	42	32.1	OX2	STDS	INNHB
-200.14	1	322	157	38	29.0	OX2	KBR	INNHB
-198.19	1	549	155	43	32.8	PMX	KBR	INNHB
-196.47	1	187	161	**59**	**45.0**	CX	KBR	INNHB
-184.53	1	696	168	38	29.0	PMX	STDS	INNHB
-190.2	1	453	167	51	38.9	CX	STDS	INN
-189.6	1	417	167	43	32.8	OX2	STDS	INN
-187.9	1	529	161	36	27.5	OX2	KBR	INN
-183.6	1	380	169	**54**	**41.2**	PMX	KBR	INN
-181.0	1	608	165	30	22.9	CX	KBR	INN
-168.0	1	385	164	31	23.7	PMX	STDS	INN

Table A.23: Single run with highest number of correctly predicted base pairs of *Acanthamoeba griffini*, regardless of free energy grouped by thermodynamic model. The known structure contains 131 base pairs.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-197.15	1	658	161	**81**	**61.8**	OX2	STDS	INNHB
-197.12	1	211	163	68	51.9	CX	STDS	INNHB
-194.56	1	409	165	65	49.6	PMX	KBR	INNHB
-188.97	1	543	156	72	55.0	CX	KBR	INNHB
-185.75	1	446	151	66	50.4	OX2	KBR	INNHB
-175.43	1	683	156	58	44.3	PMX	STDS	INNHB
-181.5	1	540	165	**75**	**57.3**	CX	STDS	INN
-181.4	1	649	167	65	49.6	OX2	KBR	INN
-180.9	1	652	162	70	53.4	OX2	STDS	INN
-175.8	1	408	163	64	48.9	CX	KBR	INN
-171.0	1	653	157	62	47.3	PMX	KBR	INN
-154.0	1	609	161	50	38.2	PMX	STDS	INN

Table A.24: *Acanthamoeba griffini*, Nussinov results. Number of known base pairs is 131.

GC:AU:GU Weights	**Predicted BP**	**Correctly Predicted BP**	**Correctly Predicted (%)**
1:1:1	215	40	30.5
3:2:1	208	37	28.2
3:2:2	214	48	36.6

Table A.25: *Acanthamoeba griffini*, *mfold* results. Number of known base pairs is 131.

mfold **ΔG (kcal / mol)**	***efn2*** **ΔG (kcal / mol)**	**Predicted BP**	**Correctly Predicted BP**	**% Correctly Predicted**
-193.0	-179.03	172	67	51.1
-192.7	-182.60	173	62	47.3
-191.8	-177.46	175	56	42.7
-190.3	-177.40	172	56	42.7
-190.3	-171.34	170	59	45.0
-189.9	-181.63	175	63	48.1
-189.6	-178.83	172	69	52.7
-188.7	**-182.67**	171	64	48.9
-188.3	-174.90	174	**95**	**72.5**
-188.2	-174.92	173	53	40.5
-187.8	-181.91	177	63	48.1
-187.4	-170.18	168	52	39.7
-187.2	-180.14	173	90	68.7
-187.0	-173.47	169	89	67.9
-186.6	-170.97	173	67	51.1
-186.3	-167.52	165	44	33.6
-184.1	-173.14	177	63	48.1

A.6 *Arthrobacter globiformis* - 123 nt

Table A.26: *Arthrobacter globiformis* details

Filename	d.5.b.A.globiformis.1.bpseq
Organism	*Arthrobacter globiformis*
Accession Number	M16173
Class	5S rRNA
Length	123 nucleotides
# of BPs in known structure	39
# of non-canonical base pairs	5

Table A.27: Results of comparison with known *Arthrobacter globiformis* structure grouped by thermodynamic model. The known structure contains 39 base pairs. Each row represents an experiment consisting of 30 averaged runs.

ΔG (kcal / mol)	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-56.94	38.0	**17.0**	**43.6**	OX2	STDS	INNHB
-56.94	38.0	**17.0**	**43.6**	CX	STDS	INNHB
-56.94	38.0	**17.0**	**43.6**	OX2	KBR	INNHB
-56.94	38.0	**17.0**	**43.6**	CX	KBR	INNHB
-56.94	38.0	**17.0**	**43.6**	PMX	KBR	INNHB
-56.94	38.0	**17.0**	**43.6**	PMX	STDS	INNHB
-54.8	38.0	**17.0**	**43.6**	OX2	STDS	INN
-54.8	38.0	**17.0**	**43.6**	CX	STDS	INN
-54.8	38.0	**17.0**	**43.6**	OX2	KBR	INN
-54.8	38.0	**17.0**	**43.6**	PMX	KBR	INN
-54.8	38.0	**17.0**	**43.6**	CX	KBR	INN
-54.8	38.0	**17.0**	**43.6**	PMX	STDS	INN

Table A.28: Best results of comparison with known *Arthrobacter globiformis* structure grouped by thermodynamic model. The known structure contains 39 base pairs. Best single run ranked by free energy.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-56.94	30	22.2	38.0	**17.0**	**43.6**	CX	KBR	INNHB
-56.94	30	22.4	38.0	**17.0**	**43.6**	CX	STDS	INNHB
-56.94	30	23.0	38.0	**17.0**	**43.6**	OX2	STDS	INNHB
-56.94	30	30.5	38.0	**17.0**	**43.6**	OX2	KBR	INNHB
-56.94	30	37.1	38.0	**17.0**	**43.6**	PMX	KBR	INNHB
-56.94	30	38.5	38.0	**17.0**	**43.6**	PMX	STDS	INNHB
-54.8	30	26.3	38.0	**17.0**	**43.6**	CX	STDS	INN
-54.8	30	29.9	38.0	**17.0**	**43.6**	OX2	STDS	INN
-54.8	30	41.2	38.0	**17.0**	**43.6**	CX	KBR	INN
-54.8	30	46.9	38.0	**17.0**	**43.6**	OX2	KBR	INN
-54.8	30	49.4	38.0	**17.0**	**43.6**	PMX	KBR	INN
-54.8	30	68.6	38.0	**17.0**	**43.6**	PMX	STDS	INN

Table A.29: Single run with highest number of correctly predicted base pairs of *Arthrobacter globiformis*, regardless of free energy grouped by thermodynamic model. The known structure contains 39 base pairs.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-56.94	30	22.4	38.0	**17.0**	**43.6**	CX	STDS	INNHB
-56.94	30	23.0	38.0	**17.0**	**43.6**	OX2	STDS	INNHB
-56.94	30	30.5	38.0	**17.0**	**43.6**	OX2	KBR	INNHB
-56.94	30	37.1	38.0	**17.0**	**43.6**	PMX	KBR	INNHB
-56.94	30	38.5	38.0	**17.0**	**43.6**	PMX	STDS	INNHB
-54.80	30	41.2	38.0	**17.0**	**43.6**	CX	KBR	INNHB
-54.8	30	26.33	38.00	**17.0**	**43.6**	CX	STDS	INN
-54.8	30	29.90	38.00	**17.0**	**43.6**	OX2	STDS	INN
-54.8	30	41.23	38.00	**17.0**	**43.6**	CX	KBR	INN
-54.8	30	46.90	38.00	**17.0**	**43.6**	OX2	KBR	INN
-54.8	30	49.43	38.00	**17.0**	**43.6**	PMX	KBR	INN
-54.8	30	68.63	38.00	**17.0**	**43.6**	PMX	STDS	INN

Table A.30: *Arthrobacter globiformis*, Nussinov results. Number of known base pairs is 39.

GC:AU:GU Weights	**Predicted BP**	**Correctly Predicted BP**	**Correctly Predicted (%)**
1:1:1	46	14	35.8
3:2:1	44	9	23.0
3:2:2	46	14	35.8

Table A.31: *Arthrobacter globiformis*, *mfold* results. Number of known base pairs is 39.

mfold **ΔG (kcal / mol)**	***efn2*** **ΔG (kcal / mol)**	**Predicted BP**	**Correctly Predicted BP**	**% Correctly Predicted**
-52.2	-46.07	37	15	38.5
-50.1	-47.01	35	15	38.5
-49.7	**-47.78**	37	**25**	**64.1**

A.7 *Aureoumbra lagunensis* - 468 nt

Table A.32: *Aureoumbra lagunensis* details

Filename	b.I1.e.A.lagunensis.C1.SSU.516.bpseq
Organism	*Aureoumbra lagunensis*
Accession Number	U40258
Class	Group I intron, 16S rRNA
Length	468 nucleotides
# of BPs in known structure	113
# of non-canonical base pairs	4

Table A.33: Results of comparison with known *Aureoumbra lagunensis* structure grouped by thermodynamic model. The known structure contains 113 base pairs. Each row represents an experiment consisting of 30 averaged runs.

ΔG (kcal / mol)	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-174.09	127.2	**41.5**	**36.7**	OX2	STDS	INNHB
-171.52	125.7	41.0	36.3	CX	STDS	INNHB
-164.18	124.7	33.6	29.7	PMX	KBR	INNHB
-163.36	123.8	34.0	30.1	OX2	KBR	INNHB
-160.81	122.2	32.9	29.1	CX	KBR	INNHB
-149.24	120.0	25.3	22.4	PMX	STDS	INNHB
-170.4	129.6	**47.1**	**41.7**	OX2	STDS	INN
-166.0	128.1	38.9	34.5	CX	STDS	INN
-157.5	124.9	33.6	29.7	PMX	KBR	INN
-157.5	126.4	29.2	25.8	OX2	KBR	INN
-156.1	124.0	27.2	24.1	CX	KBR	INN
-142.5	120.5	21.6	19.1	PMX	STDS	INN

Table A.34: Best results of comparison with known *Aureoumbra lagunensis* structure grouped by thermodynamic model. The known structure contains 113 base pairs. Best single run ranked by free energy.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-181.78	1	460	133	53	46.9	OX2	STDS	INNHB
-180.21	1	503	131	**64**	**56.6**	OX2	KBR	INNHB
-179.45	1	230	133	53	46.9	CX	KBR	INNHB
-178.92	1	534	134	58	51.3	CX	STDS	INNHB
-177.40	1	690	137	41	36.3	PMX	KBR	INNHB
-165.62	1	698	121	49	43.4	PMX	STDS	INNHB
-178.5	1	374	131	59	52.2	OX2	STDS	INN
-178.4	1	322	135	53	46.9	CX	STDS	INN
-177.8	1	358	136	60	53.1	CX	KBR	INN
-175.8	1	439	127	49	43.4	PMX	KBR	INN
-167.9	1	464	135	**65**	**57.5**	OX2	KBR	INN
-157.0	1	605	121	43	38.1	PMX	STDS	INN

Table A.35: Single run with highest number of correctly predicted base pairs of *Aureoumbra lagunensis*, regardless of free energy grouped by thermodynamic model. The known structure contains 113 base pairs.

ΔG (kcal / mol)	Freq.	Gens	Pred. BPs	Corr. BPs	Corr. BPs (%)	Cross.	Sel.	Model
-180.34	1	629	130	64	56.6	OX2	STDS	INNHB
-180.21	1	503	131	64	56.6	OX2	KBR	INNHB
-178.31	1	513	131	**68**	**60.2**	CX	STDS	INNHB
-170.32	1	690	126	55	48.7	PMX	KBR	INNHB
-168.93	1	630	128	59	52.2	CX	KBR	INNHB
-163.03	1	597	126	51	45.1	PMX	STDS	INNHB
-177.8	1	358	136	60	53.1	CX	KBR	INN
-175.8	1	422	136	60	53.1	CX	STDS	INN
-174.6	1	537	131	**66**	**58.4**	OX2	STDS	INN
-168.0	1	573	130	65	57.5	PMX	KBR	INN
-167.9	1	464	135	65	57.5	OX2	KBR	INN
-147.8	1	559	124	51	45.1	PMX	STDS	INN

Table A.36: *Aureoumbra lagunensis*, Nussinov results. Number of known base pairs is 113.

GC:AU:GU Weights	**Predicted BP**	**Correctly Predicted BP**	**Correctly Predicted (%)**
1:1:1	173	27	23.8
3:2:1	168	9	7.9
3:2:2	172	30	26.5

Table A.37: *Aureoumbra lagunensis*, *mfold* results. Number of known base pairs is 113.

mfold **ΔG (kcal / mol)**	***efn2*** **ΔG (kcal / mol)**	**Predicted BP**	**Correctly Predicted BP**	**% Correctly Predicted**
-160.1	-142.35	128	60	53.1
-159.7	**-143.71**	136	60	53.1
-158.1	-141.78	134	60	53.1
-156.6	-143.17	134	61	54.0
-156.4	-138.52	133	63	55.8
-156.2	-140.50	132	60	53.1
-155.7	-143.49	137	72	63.7
-154.5	-141.88	131	72	63.7
-154.5	-138.76	130	72	63.7
-153.9	-136.16	133	48	42.5
-153.8	-136.47	140	60	53.1
-153.8	-140.57	133	**74**	**65.5**
-153.4	-134.89	125	51	45.1
-153.3	-140.79	131	60	53.1

A.8 Over-prediction of base pairs

This section will compare the number of false positive base pairs predicted by RnaPredict, Nussinov, and *mfold* for all eleven sequences.

Table A.38: Comparison between the number of false predictions between best results with the Nussinov DPA and the best average runs with RnaPredict

Sequence	GC:AU:GU Weights	DPA over-pred.	GA over-pred.	DPA Corr. BPs	GA Corr. BPs	GA Cross.-Sel.-Model
S. acidocaldarius	3:2:2	395	335.6	187	92.0	CX-STDS-INN
H. sapiens	1:1:1	309	190.6	33	48.1	OX2-STDS-INNHB
C. elegans	3:2:2	281	172.4	26	30.9	OX2-STDS-INN
A. griffini	3:2:2	166	119.5	48	45.2	OX2-STDS-INN
A. globiformis	1:1:1	32	21.0	14	17.0	OX2-STDS-INNHB
A. lagunensis	3:2:2	142	82.5	30	47.1	OX2-STDS-INN
X. laevis	3:2:1	286	177.4	47	62.9	CX-STDS-INN
D. virilis	1:1:1	291	195.6	29	43.7	CX-STDS-INN
H. rubra	3:2:1	174	111.6	31	48.4	OX2-STDS-INNHB
H. marismortui	1:1:1	37	14.0	8	16.0	OX2-STDS-INN
S. cerevisiae	1:1:1	17	6.0	28	33.0	CX-STDS-INNHB

Table A.39: Comparison between the number of false predictions between best results with the Nussinov DPA and the single lowest energy runs with RnaPredict

Sequence	GC:AU:GU Weights	DPA over-pred.	GA over-pred.	DPA Corr. BPs	GA Corr. BPs	GA Cross.-Sel.-Model
S. acidocaldarius	3:2:2	395	308	187	131	CX-STDS-INNHB
H. sapiens	1:1:1	309	173	33	72	OX2-STDS-INNHB
C. elegans	3:2:2	281	175	26	37	OX2-STDS-INNHB
A. griffini	3:2:2	166	116	48	51	CX-STDS-INN
A. globiformis	1:1:1	32	21	14	17	OX2-STDS-INN
A. lagunensis	3:2:2	142	72	30	59	OX2-STDS-INN
X. laevis	3:2:1	286	178	47	77	OX2-STDS-INNHB
D. virilis	1:1:1	291	206	29	39	OX2-STDS-INN
H. rubra	3:2:1	174	102	31	62	OX2-STDS-INNHB
H. marismortui	1:1:1	37	14	8	16	OX2-STDS-INN
S. cerevisiae	1:1:1	17	6	28	33	OX2-STDS-INNHB

Table A.40: Comparison between the number of false predictions between best results with the Nussinov DPA and the runs predicting the highest number of known base pairs with RnaPredict

Sequence	GC:AU:GU Weights	DPA over-pred.	GA over-pred.	DPA Corr. BPs	GA Corr. BPs	GA Cross.-Sel.-Model
S. acidocaldarius	3:2:2	395	288	187	144	CX-STDS-INN
H. sapiens	1:1:1	309	161	33	89	OX2-STDS-INNHB
C. elegans	3:2:2	281	147	26	55	OX2-STDS-INN
A. griffini	3:2:2	166	80	48	81	OX2-STDS-INNHB
A. globiformis	1:1:1	32	21	14	17	CX-STDS-INNHB
A. lagunensis	3:2:2	142	63	30	68	CX-STDS-INNHB
X. laevis	3:2:1	286	147	47	93	CX-STDS-INN
D. virilis	1:1:1	291	177	29	65	OX2-STDS-INNHB
H. rubra	3:2:1	174	82	31	79	OX2-STDS-INN
H. marismortui	1:1:1	37	3	8	27	PMX-KBR-INNHB
S. cerevisiae	1:1:1	17	6	28	33	OX2-STDS-INNHB

Table A.41: Comparison between the number of false predictions between lowest energy structure found with the *mfold* DPA and the overall lowest energy single RnaPredict runs

Sequence	DPA over-pred.	GA over-pred.	DPA Corr. BPs	GA Corr. BPs	GA Cross.-Sel.-Model
S. acidocaldarius	233	308	261	131	CX-STDS-INNHB
H. sapiens	163	173	95	72	OX2-STDS-INNHB
C. elegans	177	175	40	37	OX2-STDS-INNHB
A. griffini	105	116	67	51	CX-STDS-INN
A. globiformis	22	21	15	17	OX2-STDS-INN
A. lagunensis	68	72	60	59	OX2-STDS-INN
X. laevis	157	178	92	77	OX2-STDS-INNHB
D. virilis	199	206	37	39	OX2-STDS-INN
H. rubra	127	102	49	62	OX2-STDS-INNHB
H. marismortui	5	14	29	16	OX2-STDS-INN
S. cerevisiae	8	6	33	33	OX2-STDS-INNHB

Table A.42: Comparison between the number of false predictions between best structure with the *mfold* DPA and the overall best single structure found with RnaPredict

Sequence	DPA over-pred.	GA over-pred.	DPA Corr. BPs	GA Corr. BPs	GA Cross.-Sel.-Model
S. acidocaldarius	225	288	271	144	CX-STDS-INN
H. sapiens	163	161	95	89	OX2-STDS-INNHB
C. elegans	177	147	40	55	OX2-STDS-INN
A. griffini	79	80	95	81	OX2-STDS-INNHB
A. globiformis	12	21	25	17	CX-STDS-INNHB
A. lagunensis	59	63	74	68	CX-STDS-INNHB
X. laevis	132	147	113	93	CX-STDS-INN
D. virilis	170	177	82	65	OX2-STDS-INNHB
H. rubra	84	82	83	79	OX2-STDS-INN
H. marismortui	5	3	29	27	PMX-KBR-INNHB
S. cerevisiae	8	6	33	33	OX2-STDS-INNHB

Bibliography

[1] Kay C. Wiese and Edward Glen. A permutation based genetic algorithm for RNA secondary structure prediction. In Ajith Abraham, Javier Ruiz del Solar, and Mario Koppen, editors, *Soft Computing Systems*, volume 87 of *Frontiers in Artificial Intelligence and Applications*, chapter 4, pages 173–182. IOS Press, Amsterdam, 2002.

[2] A. Deschênes, K. C. Wiese, and Jagdeep Poonian. Comparison of dynamic programming and evolutionary algorithms for RNA secondary structure prediction. In *Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB'04)*, pages 214–222. IEEE Press, Oct 2004.

[3] Kyungsook Han, Yujin Lee, and Wootaek Kim. Pseudoviewer: automatic visualization of RNA pseudoknots. *Bioinformatics*, 18:S321–S328, March 2002.

[4] Ruth Nussinov, George Pieczenik, Jerrold R. Griggs, and Daniel J. Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied Mathematics*, 35:68–82, 1978.

[5] Michael Zuker and Patrick Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9:133–148, 1981.

[6] Michael Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, 1989.

[7] Michael Zuker. Prediction of RNA secondary structure by energy minimization. In Annette M. Griffin and Hugh G. Griffin, editors, *Computer Analysis of Sequence Data*, pages 267–294. Humana Press Inc., July 1994.

[8] M. Zuker, D. H. Mathews, and D. H. Turner. Algorithms and thermodynamics for RNA secondary structure prediction: a practical guide. In J. Barciszewski and B.F.C. Clark, editors, *RNA Biochemistry and Biotechnology*, NATO ASI Series. Kluwer Academic Publishers, 1999.

[9] Michael Zuker. Calculating nucleic acid secondary structure. *Current Opinion in Structural Biology*, 10:303–310, 2000.

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

[11] D. H. Mathews, M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker, and D. H. Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences, USA*, 101:7287–7292, 2004.

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

[13] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6–7):1105–1119, 1990.

[14] Martin Fekete, Ivo L. Hofacker, and Peter F. Stadler. Prediction of RNA base pairing probabilities on massively parallel computers. *Journal of Computational Biology*, 7(1/2):171–182, 2000.

[15] Stefan Wuchty. Suboptimal secondary structures of RNA. Master's thesis, University of Vienna, 1998.

[16] S. Wuchty, W. Fontana, I. Hofacker, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49:145–165, 1999.

[17] Christoph Flamm, Ivo L. Hofacker, and Peter F. Stadler. RNA in silico the computational biology of RNA secondary structures. *Advances in Complex Systems*, 1:65–90, 1999.

[18] Christoph Flamm, Walter Fontana, Ivo L. Hofacker, and Peter Schuster. RNA folding at elementary step resolution. *RNA*, 6:325–338, 2000.

[19] D. Sankoff. Simultaneous solution of the RNA folding, alignment, and proto-sequence problems. *SIAM Journal on Applied Mathematics*, 45:810–825, 1985.

[20] C. R. Woese and N. R. Pace. Probing RNA structure, function and history by comparative analysis. In R. F. Gesteland and J. F. Atkins, editors, *The RNA World.* Cold Spring Harbor, NY, 1993.

[21] J. Gorodkin, L. Heyer, and G. Stormo. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Research*, 25(18):3724–3732, 1997.

[22] J. Gorodkin, S. L. Stricklin, and G. D. Stormo. Discovering common stemloop motifs in unaligned RNA sequences. *Nucleic Acids Research*, 29(10):2135–2144, 2001.

[23] David H. Mathews and Douglas H. Turner. Dynalign: An algorithm for finding the secondary structure common to two RNA sequences. *Journal of Molecular Biology*, 317:191–203, 2002.

[24] Ivo L. Hofacker, Stephan H. F. Bernhart, and Peter F. Stadler. Alignment of RNA base pairing probability matrices. *Bioinformatics*, 20(14):2222–2227, 2004.

[25] Paul P. Gardner and Robert Giegerich. A comprehensive comparison of comparative RNA structure prediction approaches. *BMC Bioinformatics*, 5(140), 2004.

[26] F. H. D. van Batenburg, A. P. Gultyaev, and C. W. A. Pleij. The computer simulation of RNA folding pathways using a genetic algorithm. *Journal of Molecular Biology*, 250:37–51, 1995.

[27] F. H. D. van Batenburg, Alexander P. Gultyaev, and Cornelis W. A. Pleij. An APL-programmed genetic algorithm for the prediction of RNA secondary structure. *Journal of Theoretical Biology*, 174:269–280, 1995.

[28] B. A. Shapiro and J. Navetta. A massively-parallel genetic algorithm for RNA secondary structure prediction. *Journal of Supercomputing*, 8:195–207, 1994.

[29] G. Benedetti and S. Morosetti. A genetic algorithm to search for optimal and suboptimal RNA secondary structures. *Biophysical Chemistry*, 55:253–259, 1995.

[30] Alexander P. Gultyaev, F. H. D. van Batenburg, and Cornelis W. A. Pleij. The computer-simulation of RNA folding pathways using a genetic algorithm. *Journal of Molecular Biology*, 250:37–51, 1995.

[31] Bruce A. Shapiro, David Bengali, and Wojciech Kasprzak. Determination of RNA folding pathway functional intermediates using a massively parallel genetic algorithm. Abstract and references.

[32] B. A. Shapiro, J. C. Wu, D. Bengali, and M. J. Potts. The massively parallel genetic algorithm for RNA folding: Mimd implementation and population variation. *Bioinformatics*, 17:137–148, 2001.

[33] B. A. Shapiro and J. C. Wu. An annealing mutation operator in the genetic algorithms for RNA folding. *Computer Applications in the Biosciences*, 12:171–180, 1996.

[34] Kay C. Wiese and Edward Glen. A permutation-based genetic algorithm for the RNA folding problem: a critical look at selection strategies, crossover operators, and representation issues. *BioSystems - Special Issue on Computational Intelligence in Bioinformatics*, 72:29–41, 2003.

[35] Kay C. Wiese, Alain Deschênes, and Edward Glen. Permutation based RNA secondary structure prediction via a genetic algorithm. In Ruhul Sarker, Robert Reynolds, Hussein Abbass, Kay Chen Tan, Bob McKay, Daryl Essam, and Tom Gedeon, editors, *Proceedings of the 2003 Congress on Evolutionary Computation (CEC2003)*, pages 335–342, Canberra, 8-12 December 2003. IEEE Press.

[36] A. Deschênes, K. C. Wiese, and E. Glen. Comparison of permutation-based and binary representation in a genetic algorithm for RNA secondary structure prediction. In A. Y. Tawfik and S. D. Goodwin, editors, *Advances in Artificial Intelligence, 17th Conference of the Canadian Society for Computational Studies of Intelligence*, volume

3060 of *LNAI*, pages 549–550, London, Ontario, Canada, May 2004. Canadian AI, Springer.

[37] Alain Deschênes and Kay C. Wiese. Using stacking-energies (INN and INN-HB) for improving the accuracy of RNA secondary structure prediction with an evolutionary algorithm - a comparison to known structures. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, volume 1, pages 598–606, Portland, Oregon, Jun 2004. IEEE Press.

[38] Andrew Hendriks, Kay C. Wiese, Edward Glen, and Alain Deschênes. A distributed genetic algorithm for RNA secondary structure prediction. In Ruhul Sarker, Robert Reynolds, Hussein Abbass, Kay Chen Tan, Bob McKay, Daryl Essam, and Tom Gedeon, editors, *Proceedings of the 2003 Congress on Evolutionary Computation (CEC2003)*, pages 343–350. IEEE Press, Dec 2003.

[39] A. Hendriks, A. Deschênes, and K. C. Wiese. A parallel evolutionary algorithm for RNA secondary structure prediction using stacking-energies (INN and INN-HB). In *Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB'04)*, pages 223–230. IEEE Press, Oct 2004.

[40] Albert L. Lehninger, David L. Nelson, and Michael M. Cox. *Lehninger Principles of Biochemistry*. W.H. Freeman & Company, 4^{th} edition, 2004.

[41] P. G. Higgs. RNA secondary structure: physical and computational aspects. *Quarterly Reviews of Biophysics*, 33:199–253, 2000.

[42] Gerald F. Joyce. Evolution of catalytic function. *Pure and Applied Chemistry*, 65(6):1205–1212, 1993.

[43] Jennifer A. Doudna and Thomas R. Cech. The chemical repertoire of natural ribozymes. *Nature*, 418:222–228, July 2002. Review.

[44] Qi Chen, Richard H. Shafer, and Irwin D. Kuntz. Structure-based discovery of ligands targeted to the RNA double helix. *Biochemistry*, 36(38):11402–11407, 1997.

[45] R. W. Holley, J. Apgar, G. A. Everett, J. T. Madison, M. Marquise, S. H. Merrill, J. R. Penswick, and A. Zamir. Structure of a ribonucleic acid. *Science*, 147:1462–1465, 1965.

[46] Gabriele Varani, Fareed Aboul-ela, and Frederic H. T. Allain. NMR investigation of RNA structure. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 29(1-2):51–127, June 1996.

[47] B. Fürtig, C. Richter, J. Wöhnert, and H. Schwalbe. NMR spectroscopy of RNA. *Chembiochem*, 4(10):936–962, 2003.

[48] Stephen R. Holbrook and Sung-Hou Kim. RNA crystallography. *Biopolymers*, 44(1):3–21, 1997.

[49] P. N. Borer, B. Dengler, I. Tinoco Jr., and O. C. Uhlenbeck. Stability of ribonucleic acid double-stranded helices. *Journal of Molecular Biology*, 86:843–853, 1974.

[50] John A. Jaeger, Douglas H. Turner, and Michael Zuker. Improved predictions of secondary structures for RNA. *Biochemistry*, 86:7706–7710, October 1989.

[51] Michael Zuker, John A. Jaeger, and Douglas H. Turner. A comparison of optimal and suboptimal RNA secondary structures predicted by free energy minimization with structures determined by phylogenetic comparison. *Nucleic Acids Research*, 19(10):2707–2714, 1991.

[52] D. H. Mathews, T. C. Andre, J. Kim, D. H. Turner, and M. Zuker. An updated recursive algorithm for RNA secondary structure prediction with improved free energy parameters. In N. B. Leontis and J. SantaLucia Jr., editors, *American Chemical Society*, 682, chapter 15, pages 246–257. American Chemical Society, Washington, DC, 1998.

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

[54] Susan M. Freier, Barbara J. Burger, Dirk Alkema, Thomas Neilson, and Douglas H. Turner. Effects of 3' dangling end stacking on the stability of GGCC and CCGG double helixes. *Biochemistry*, 22(26):6198–6206, 1983.

[55] Ryszard Kierzek, Marvin H. Caruthers, Carl E. Longfellow, David Swinton, Douglas H. Turner, and Susan M. Freier. Polymer-supported RNA synthesis and its application to test the nearest neighbor model for duplex stability. *Biochemistry*, 25:7840–7846, June 1986.

[56] S. M. Freier, R. Kierzek, J. A. Jaeger, N. Sugimoto, M. H. Caruthers, T. Neilson, and D. H. Turner. Improved free-energy parameters for predictions of RNA duplex stability. *Proceedings of the National Academy of Sciences of the United States of America*, 83:9373–9377, 1986.

[57] Naoki Sugimoto, Ryszard Kierzek, Susan M. Freier, and Douglas H. Turner. Energetics of internal GU mismatches in ribooligonucleotide helixes. *Biochemistry*, 25(19):5755 – 5759, 1986.

[58] Liyan He, Ryszard Kierzek, Jr. John SantaLucia, Amy E. Walter, and Douglas H. Runer. Nearest-neighbor parameters for GU mismatches: GU/UG is destabilizing in the contexts CGUG/GUGC, UGUA/AUGU but stabillizing in GGUC/CUGG. *Biochemistry*, 30:11124–11132, 1991.

[59] Ming Wu, Jeffrey A. McDowell, and Douglas H. Turner. A periodic table of symmetric tandem mismatches in RNA. *Biochemistry*, 34:3204–3211, 1995.

[60] Martin J. Serra and Douglas H. Turner. Predicting thermodynamic properties of RNA. *Methods in Enzymology*, 259:242–261, 1995.

[61] Tianbing Xia, Jeffrey A. McDowell, and Douglas H. Turner. Thermodynamics of nonsymmetric tandem mismatches adjacent to GC base pairs in RNA. *Biochemistry*, 36:12486–12497, 1997.

[62] Tianbing Xia, John SantaLucia Jr., Mark E. Burkard, Ryszard Kierzek, Susan J. Schroeder, Xiaoqi Jiao, Christopher Cox, and Douglas H. Turner. Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry*, 37:14719–14735, 1998.

[63] I. Tinoco, O. Uhlenbeck, and M. D. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, 1971.

[64] R. Fink and D. M. Crothers. Free energy of imperfect nucleic acis helices. I. The bulge defect. *Journal of Molecular Biology*, 66:1–12, 1972.

[65] C. E. Longfellow, R. Kierzek, and D. H. Turner. Thermodynamic and spectroscopic study of bulge loops in oligoribonucleotides. *Biochemistry*, 29(1):278–285, Jan 1990.

[66] J. Gralla and D. M. Crothers. Free energy of imperfect nucleic acid helices. III. Small internal loops resulting from mismatches. *Journal of Molecular Biology*, 78(2):301–319, 1973.

[67] R. Kierzek, M. E. Burkard, and D. H. Turner. Thermodynamics of single mismatches in RNA duplexes. *Biochemistry*, 38:14214–14223, 1999.

[68] May Meroueh and Christine S. Chow. Thermodynamics of RNA hairpins containing single internal mismatches. *Nucleic Acids Research*, 27(4), 1999.

[69] Amy E. Walter, Ming Wu, and Douglas H. Turner. The stability and structure of tandem GA mismatches in RNA depend on closing base pairs. *Biochemistry*, 33:11349–11354, 1994.

[70] Susan Schroeder, James Kim, and Douglas H. Turner. GA and UU mismatches can stabilize RNA internal loops of three nucleotides. *Biochemistry*, 35:16105–16109, 1996.

[71] Susan J. Schroeder and Douglas H. Turner. Factors affecting the thermodynamic stability of small asymmetric internal loops in RNA. *Biochemistry*, 39:9257–9274, 2000.

[72] Susan J. Schroeder and Douglas H. Turner. Thermodynamic stabilities of internal loops with GU closing pairs in RNA. *Biochemistry*, 40:11509–11517, 2001.

[73] O. C. Uhlenbeck, P. N. Borer, B. Dengler, and I. Tinoco Jr. Stability of RNA hairpin loops: A_6-c_m-u_6. *Journal of Molecular Biology*, 33:483–496, 1973.

[74] J. Gralla and D. M. Crothers. Free energy of imperfect nucleic acid helices II. Small hairpin loops. *Journal of Molecular Biology*, 73:497–511, 1973.

[75] D. R. Groebe and O. C. Uhlenbeck. Characterization of RNA hairpin loop stability. *Nucleic Acids Research*, 16:11725–11735, 1988.

[76] Martin J. Serra, Thomas W. Barnes, Kelly Betschart, Mathew J. Gutierrez, Kimberly J. Sprouse, Cheryl K. Riley, Lora Stewart, and Ryan E. Temel. Improved parameters for the prediction of RNA hairpin stability. *Biochemistry*, 36:4844–4851, 1997.

[77] Matthew R. Giese, Kelly Betschart, Taraka Dale, Cheryl K. Riley, Carrie Rowan, Kimberly J. Sprouse, and Martin J. Serra. Stability of RNA hairpins closed by wobble base pairs. *Biochemistry*, 37:1094–1100, 1998.

[78] Zhanyong Shu and Philip C. Bevilacqua. Isolation and characterization of thermodynamically stable and unstable RNA hairpins from a triloop combinatorial library. *Biochemistry*, 38:15369–15379, 1999.

[79] Craig Tuerk, Peter Gauss, Claude Thermes, Duncan R. Groebe, Margit Gayle, Nancy GUild, Gary Stormo, Yves D'aubenton-carafa, Olke C. Uhlenbeck, Jr. Ignacio Tinoko, Edward N. Brody, and Larry Gold. CUUCGG hairpins: Extraordinarily stable RNA secondary structures associated with various biochemical processes. *Biochemistry*, 85:1364–1368, October 1988.

[80] Duncan R. Groebe and Olke C . Uhlenbeck. Thermal stability of RNA hairpins containing a four-membered loop and a bulge nucleotide. *Biochemistry*, 28:742–747, 1989.

[81] D. R. Groebe and O. C. Uhlenbeck. Thermal stability of RNA hairpins containing a four-membered loop and a bulge nucleotide. *Biochemistry*, 28:742–747, 1989.

[82] O. C. Uhlenbeck. Tetraloops and RNA folding. *Nature*, 346:613–614, 1990.

[83] Gabriele Varani, Chaejoon Cheong, and Jr. Ignacio Tinoco. Structure of an unusually stable RNA hairpin. *Biochemistry*, 30:3280–3289, 1991.

[84] V. P. Antao, S. Y. Lai, and Jr. I. Tinoco. A thermodynamic study of unusually stable RNA and DNA hairpins. *Nucleic Acids Research*, 19:5901–5905, 1991.

[85] Vincent P. Antao and Jr. Ignacio Tinoco. Thermodynamic parameters for loop formation in RNA and DNA hairpin tetraloops. *Nucleic Acids Research*, 20(4), January 1992.

[86] Martin J. Serra, Matthew H. Lyttle, Theresa J. Axenson, Calvin A. Schadt, and Douglas H. Turner. RNA hairpin loop stability depends on closing base pair. *Nucleic Acids Research*, 21(16):3845–3849, 1993.

[87] Martin J. Serra, Theresa J. Axenson, and Douglas H. Turner. A model for the stabilities of RNA hairpins based on a study of the sequence dependence of stability for hairpins of six nucleotides. *Biochemistry*, 33:14289–14296, 1994.

[88] J. M. Diamond, D. H. Turner, and D. H. Mathews. Thermodynamics of three-way multibranch loops in RNA. *Biochemistry*, 40:6971–6981, 2001.

[89] David H. Mathews and Douglas H. Turner. Experimentally derived nearest-neighbor parameters for the stability of RNA three- and four-way multibranch loops. *Biochemistry*, 41:869–880, 2002.

[90] Alexander P. Gultyaev, F. H. D. van Batenburg, and Cornelis W. A. Pleij. An approximation of loop free energy values of RNA H-pseudoknots. *RNA*, 5:609–617, 1999. Describes H-Type pseudoknots and gives an example. Gives thermodynamic model and relates it to a known struture.

[91] Rune B. Lyngso and Christian N. S. Pedersen. RNA pseudoknot prediction in energy-based models. *Journal of Computational Biology*, 7(3/4):409–427, 2000.

[92] Rune B. Lyngso and Christian N. S. Pedersen. Pseudoknots in RNA secondary structures. In *Proceedings of the fourth annual international conference on Computational molecular biology*, pages 201–209. ACM Press, 2000.

[93] Amy E. Walter and Douglas H. Turner. Sequence dependence of stability for coaxial stacking of RNA helixes with Watson-Crick base paired interfaces. *Biochemistry*, 33:12715–12719, 1994.

[94] A. Walter, D. Turner, J. Kim, M. Lyttle, P. Moller, D. Mathews, and M. Zuker. Coaxial stacking of helices enhances binding of oligoribonucleotides and improves predictions of RNA folding. *Proceedings of the National Academy of Sciences*, 91:9218–9222, 1994.

[95] Amy E. Walter and Douglas H. Turner. Sequence dependence of stability for coaxial stacking of RNA helixes with watson-crick base paired interfaces. *Biochemistry*, 33:12715–12719, 1994.

[96] James Kim, Amy E. Walter, and Douglas H. Turner. Thermodynamics of coaxially stacked helixes with GA and CC mismatches. *Biochemistry*, 35:13753–13761, 1996.

[97] Jamie J Cannone, Sankar Subramanian, Murray N Schnare, James R Collett, Lisa M D'Souza, Yushi Du, Brian Feng, Nan Lin, Lakshmi V Madabusi, Kirsten M Müller, Nupur Pande, Zhidi Shang, Nan Yu, and Robin R Gutell. The comparative RNA web

(CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics*, 3, 2002.

[98] Eric W. Weisstein. Correlation coefficient. `http://mathworld.wolfram.com/CorrelationCoefficient.html`. From MathWorld–A Wolfram Web Resource.

[99] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

[100] Thomas Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms.* Oxford University Press, 1996.

[101] Bruce A. Shapiro, David Bengali, Wojciech Kasprzak, and Jin Chu Wu. RNA folding pathway functional intermediates: Their prediction and analysis. *Journal of Molecular Biology*, 312:27–44, 2001.

[102] J. H. Chen, S. Y. Le, and J. V. Maizel. Prediction of common secondary structures of RNAs: a genetic algorithm approach. *Nucleic Acids Research*, 28:991–999, 2000.

[103] J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In J. J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms and their Application*, pages 14–21, Hillsdale, New Jersey, USA, 1987. Lawrence Erlbaum Associates.

[104] Kay Wiese and Scott D. Goodwin. Keep-best reproduction: A selection strategy for genetic algorithms. In *Proceedings of the 1998 ACM Symposium on Applied Computing*, pages 343–348. ACM, 1998.

[105] Kay Wiese. *On Genetic Selection and Sequencing.* PhD thesis, University of Regina, 1999.

[106] Kay Wiese and Scott D. Goodwin. Convergence characteristics of keep-best reproduction. In *SAC '99. Proceedings of the 1999 ACM Symposium on Applied Computing 1999*, pages 312–318. ACM, 1999.

[107] Kay Wiese and Scott D. Goodwin. Keep-best reproduction: A local family competition selection strategy and the environment it flourishes in. *Constraints*, 6(4):399–422, 2001.

[108] Lawrence Davis. Job shop scheduling with genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 136–140. Lawrence Erlbaum Associates, Inc., 1985.

[109] G. Syswerda. Handbook of genetic algorithms. In L. Davis, editor, *Handbook of Genetic Algorithms*, chapter Schedule optimization using genetic algorithms. Van Nostrand Reinhold, New York, 1991.

[110] D.E. Goldberg and Jr.R. Lingle. Alleles, loci and the travelling salesman problem. In J.J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms*, pages 154–159. Lawrence Erlbaum Associates, 1985.

[111] I. M. Oliver, D. J. Smith, and J. R. C. Holland. A study of permutation crossover operators on the traveling salesman problem. In *Proceedings of the Second International Conference on Genetic Algorithms (ICGA-87)*, pages 224–230. Lawrence Erlbaum Associates, Inc., 1987.

[112] Darrell Whitley, Timothy Starkweather, and Daniel Shaner. The traveling salesman and sequence scheduling: Quality solutions using genetic edge recombination. In Lawrence Davis, editor, *Handbook of Genetic Algorithms*, pages 350–372. Van Nostrand Reinhold, New York, 1991.

[113] Kay C. Wiese, Scott D. Goodwin, and Sivakumar Nagarajan. ASERC - a genetic sequencing operator for asymmetric permutation problems. In H. Hamilton and Q. Yang, editors, *Canadian AI 2000, LNAI 1822*, pages 201–213. Springer-Verlag Berlin Heidelberg, 2000.

[114] T. Starkweather, S. McDaniel, K. Mathias, D. Whitley, and C. Whitley. A comparison of genetic sequencing operators. In Rick Belew and Lashon Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 69–76, San Mateo, CA, 1991. Morgan Kaufman.

[115] K. De Jong. *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, Ann Arbor, MI, 1975.

[116] Bart Rylander. *Computational complexity and the genetic algorithm.* PhD thesis, University of Idaho, November 2001.

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

[118] Amnon Barak, Oren La'adan, and Amnon Shiloh. Scalable cluster computing with mosix for linux. For the University of Jerusalem, 1999.

[119] M. E. Burkard, T. B. Xia, and D. H. Turner. Thermodynamics of RNA internal loops with a guanosine-guanosine pair adjacent to another noncanonical pair. *Biochemistry*, 40:2478–2483, 2001.

[120] Sébastien Lemieux and François Major. RNA canonical and non-canonical base pairing types: a recognition method and complete repertoire. *Nucleic Acids Research*, 30(19):4258–4263, 2002.

[121] Uma Nagaswamy, Maia Larios-Sanz, James Hury, Shakaala Collins, Zhengdong Zhang, Qin Zhao, and George E. Fox. NCIR: a database of non-canonical interactions in known RNA stuctures. *Nucleic Acids Research*, 30(1):395–397, 2002.

[122] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann Publishers, San Mateo, California, USA, 1991.

[123] Kyungsook Han and Yanga Byun. Pseudoviewer 2: visualization of RNA pseudoknots of any type. *Nucleic Acids Research*, 31(13), 2003.

[124] Allison Waugh, Patrick Gendron, Russ Altman, James W. Brown, David Case, Daniel Gautheret, Stephen C. Harvey, Neocles Leontis, John Westbrook, Eric Westhof, Michael Zuker, and Francois Major. RNAml: A standard syntax for exchanging RNA information. *RNA*, 8:707–717, 2002.

[125] L. Davis, editor. *A Handbook Of Genetic Algorithms.* Int. Thomson Computer Press, 1991.

[126] H. Mühlenbein, M. G. Schleuter, and O. Krämer. Evolution algorithms in combinatorial optimization. *Parallel Computing*, 1988.

[127] J. L. Blanton and R. L. Wainwright. Multiple vehicle routing with time and capacity constraints using genetic algorithms. In S. Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 452–459, San Mateo, CA, 1993. Morgan Kaufman.