Using SetPSO to determine RNA secondary structure

by

Charles Marais Neethling

Submitted in partial fulfilment of the requirements for the degree of
Master of Science (Computer Science)
in the Faculty of Engineering, Built Environment and Information Technology
University of Pretoria, Pretoria

August 2008

A research publication by

C I R G

Computational Intelligence Research Group

Visit the research group online at
`cirg.cs.up.ac.za`

Using SetPSO to determine RNA secondary structure

by

Charles Marais Neethling

Abstract

RNA secondary structure prediction is an important field in Bioinformatics. A number of different approaches have been developed to simplify the determination of RNA molecule structures. RNA is a nucleic acid found in living organisms which fulfils a number of important roles in living cells. Knowledge of its structure is crucial in the understanding of its function. Determining RNA secondary structure computationally, rather than by physical means, has the advantage of being a quicker and cheaper method. This dissertation introduces a new Set-based Particle Swarm Optimisation algorithm, known as SetPSO for short, to optimise the structure of an RNA molecule, using an advanced thermodynamic model. Structure prediction is modelled as an energy minimisation problem.

Particle swarm optimisation is a simple but effective stochastic optimisation technique developed by Kennedy and Eberhart [55]. This simple technique was adapted to work with variable length particles which consist of a set of elements rather than a vector of real numbers. The effectiveness of this structure prediction approach was compared to that of a dynamic programming algorithm called *mfold*. It was found that SetPSO can be used as a combinatorial optimisation technique which can be applied to the problem of RNA secondary structure prediction. This research also included an investigation into the behaviour of the new SetPSO optimisation algorithm. Further study needs to be conducted to evaluate the performance of SetPSO on different combinatorial and set-based optimisation problems.

Keywords: combinatorial, computational intelligence, particle swarm optimiser, RNA, secondary structure, SetPSO.

Supervisor : Prof AP Engelbrecht

Department : Department of Computer Science

Degree : Magister Scientiae

To my parents...

Acknowledgements

I would like to express my sincere thanks to the following people for their assistance during the production of this dissertation:

- Professor AP Engelbrecht, my supervisor, for his insight, encouragement and guidance throughout the work.
- Nelis Franken, for his insight and discussions around SetPSO as well as the excellent FluxViz software.
- Gary Pampara, for assisting me with queries on Linux and the CILib library.
- All colleagues and friends who took an interest in my work and encouraged me.
- My parents Hannes and Hanlie for their immense support and love.
- Last but not least, my wife Elana, who supported me graciously through this effort. Thanks for your love and for taking care of me.

This dissertation was produced with the aid of the following open source and freeware software tools. Special thanks to the authors of these superb software packages:

Typesetting using LaTeX 2_ε; Bibliographic references maintained using BibTeX; RNA structure visualisation using jViz.RNA [103]; Parallel coordinate visualisation using FluxViz [30]; CIRG's CILib computational intelligence library [14].

Contents

List of Figures

List of Graphs

List of Algorithms

List of Tables

Chapter 1

Introduction

1.1 Rationale

The first attempts to predict the structures of RNA molecules, given their nucleotide sequences, were made in 1978 by Nussinov and co-workers [70]. They tried to maximise the number of paired bases by using a dynamic programming algorithm (DPA) to optimise the RNA secondary structure. In time, the objective function used for these types of optimisation problems became more sophisticated. Nussinov *et al* published an adapted algorithm which used a simple nearest-neighbour energy model to evaluate the structures [69]. Following on that, Zuker and Stiegler proposed a more sophisticated stacking energy model in 1981 [117]. The energy parameters used in the stacking model were derived from empirical calorimetric experiments done in laboratories. Many single-structure prediction approaches still use the stacking energy model, which was last updated with the latest parameters in 2004 [64].

The dynamic programming algorithms used for these structure predictions are computationally expensive and can consume a large amount of time for predictions on longer sequences. Finding a more efficient way to predict low energy structures (the natural conformations are believed to be close to the lowest energy structures of a molecule) is high on the agenda. Several stochastic optimisation methods have been tried, most notably the genetic algorithm (GA). The approaches that use a GA to minimise the free energy of RNA structures include a massively parallel GA [84], a program by Wiese *et*

al called RnaPredict [104] [106], a GA using an annealing mutation operator [85] and various older GA approaches [97] [2]. Another approach includes CONTRAfold, which is based on stochastic context-free grammars (SCFGs). SCFGs use fully-automated statistical learning algorithms to derive model parameters and does not rely on a physical RNA thermodynamic model.

Particle swarm optimisation (PSO) has proved to be an effective yet simple optimisation algorithm [22]. Introduced in 1995 by Kennedy and Eberhart [55], PSO has made rapid advances in a number of optimisation fields and has proved to outclass other optimisation algorithms, like the GA, in many ways [24]. The original PSO is not suited to the combinatorial type of optimisation required for RNA secondary structure predictions due to its operation in continuous real vector space. This work develops a new set-based particle swarm adaptation which can be used for combinatorial optimisation problems or for finding optimal sets of elements, given an objective function. RNA structure prediction is an ideal domain for validating the modified particle swarm optimisation technique introduced in this work. This study aims to evaluate and validate the new SetPSO algorithm on a real world problem.

1.2 Objectives

The primary objectives of this dissertation are summarised as follows:

- To validate the SetPSO algorithm and make sure that it is able to optimise set-based problems.
- To implement a SetPSO algorithm in order to do RNA secondary structure predictions.
- To investigate the performance of the SetPSO when applied to the prediction of RNA secondary structure.
- To determine whether SetPSO is a viable stochastic optimisation algorithm to be used in the prediction of RNA secondary structures.

- To investigate the behaviour of the newly introduced SetPSO and the influence of its parameters on performance.
- To identify any relations between the new parameters introduced by SetPSO.

1.3 Contributions

The novel contributions of this work include the following:

- A novel adaptation of the particle swarm optimiser algorithm that can be used to optimise variable length, set-based solutions.
- A stochastic RNA secondary structure prediction method based on SetPSO and using the Vienna RNA stacking energy model as the objective function.
- The first application of particle swarm optimisation on RNA secondary structure prediction.
- An investigation of the behaviour of SetPSO when used as an optimisation algorithm.

1.4 Dissertation outline

All cited sources within this dissertation are listed in a bibliography. References to online material are only provided in instances where a document's primary publication method is online, or where a persistent Digital Object Identifier (DOI)[1] to an electronic document, provided within the CrossRef[2] framework, is available.

[1] A DOI is a *unique* alphanumeric identification string for a digital object, which provides a *persistent* link to the object in question. A DOI is a permanent URL maintained in the same way as a domain name. For further information on the DOI system, refer to `http://www.doi.org`.

[2] CrossRef is a non-profit network providing an infrastructure for linking online citations, using the DOIs of electronically published documents. A CrossRef DOI has the form `doi:10.1234/5678`. CrossRef DOIs function like standard hyperlinks in most web browsers, but may also be resolved via `http://dx.doi.org`. For information on the CrossRef system, see `http://www.crossref.org`.

The organisation of the remaining chapters of this dissertation is reflected below along with a brief description of the topics dealt within each:

- **Chapter 2** introduces the RNA molecule and describes its significance along with the various levels of RNA structure.
- **Chapter 3** gives a background view of the RNA thermodynamic models in use in various prediction methods.
- **Chapter 4** gives an overview of the original particle swarm optimiser algorithm. The algorithm and its control parameters are discussed.
- **Chapter 5** introduces the adapted PSO, i.e. SetPSO, along with the new operators that are defined to work on the set solution space.
- **Chapter 8** describes the experimental approach followed to set up SetPSO in order to predict RNA secondary structure.
- **Chapter 9** shows the preliminary results obtained from experiments run on the RNA sequences.
- **Chapter 10** contains conclusion remarks and suggestions for future work stemming from the research done in this dissertation.
- **Appendix A** provides result summaries for all the experiments done in this dissertation.
- **Appendix B** provides a list of the acronyms used and defined in this work, as well as their associated definitions.
- **Appendix C** lists and defines the mathematical symbols used in this work, categorised according to the relevant chapter in which they appear.
- **Appendix D** lists the publications derived from this work.

Chapter 2

Ribonucleic acid

Contemporary research is increasingly revealing that RNA molecules are of great importance in biological organisms. RNA is a fundamental building block of all living cells and has a variety of different functions [1] [27] [57] [71] [73] [80]. This chapter introduces ribonucleic acid and its structure. Section 2.1 introduces the basic building blocks of RNA. Then section 2.2 explains how the RNA strands are formed. Sections 2.3 and 2.4 describe RNA and its function. Sections 2.5, 2.6 and 2.7 discuss RNA primary, secondary and tertiary structure respectively and section 2.8 introduces the methods of determining RNA molecules' natural conformations.

2.1 Nucleotides

RNA molecules consist of nucleotides connected together on a sugar-phosphate backbone to form a polymer. A nucleotide is formed by the sugar-phosphate and a nitrogenous heterocyclic functional group, called a base, either a *purine* or a *pyrimidine*. The nitrogenous bases in RNA nucleotides are *adenine, cytosine, guanine* and *uracil* (A, C, G or U) (see figure 2.1). These bases are formed by hydrogen, carbon, nitrogen and oxygen molecules [47] [102].

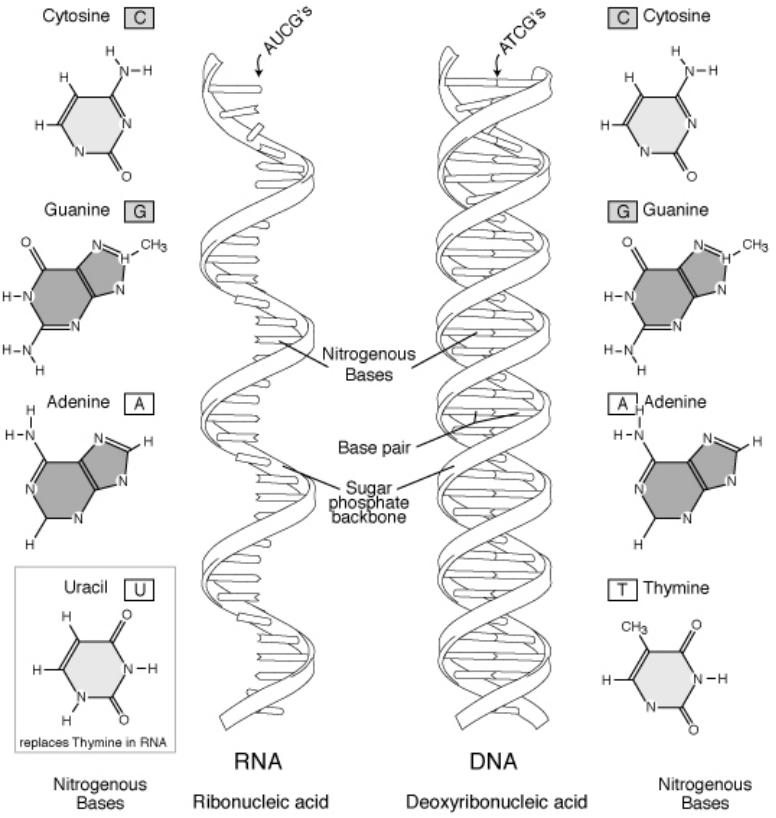


Figure 2.1: A representation of the helical structure of RNA and DNA, showing the structures of the nucleotides.

2.2 RNA strands

The nucleotides are connected together in a chain through shared oxygen atoms to form the polymer. Conventionally, the carbons to which the phosphate group is connected are called the 5' and 3' carbons. Ligation takes place between the oxygen on the 3'-hydroxyl and the oxygen on the 5'-phosphate on the ribose sugar. The 5' and 3' naming convention also indicates direction. RNA strands are always synthesised (built) in the 5' to 3' direction. This happens fairly rapidly at a rate of 50-100 nucleotides per second.

A nucleic acid that is less than 50 nucleotides long is called an oligonucleotide. Anything longer (chain with up to several thousand nucleotides) is called a polynucleotide. The lengths of RNA strands differ, depending on their respective function and classes.

2.3 RNA

RNA is a nucleic acid. This is the name given to a high molecular weight macro molecule consisting of a chain of nucleotides. The most common nucleic acids found in living cells are DNA (deoxyribonucleic acid) and RNA (ribonucleic acid). These nucleic acids convey genetic information in addition to performing many other functions, which are discussed later in this chapter.

RNA, unlike DNA, exists as a single strand (see figure 2.1). The strand then folds back onto itself. Hydrogen bonds form between certain base pairs, which are known as canonical pairs. These pairs are said to be complementary and form the strongest bonds. In RNA, *cytosine* is complementary to *guanine* where three hydrogen bonds form and *adenine* is complementary to *uracil* where two hydrogen bonds form. These are called the Watson-Crick pairs, after the discoverers of the DNA structure.

Another important base pair to consider is the so-called wobble pair, a double hydrogen bond between *guanine* and *uracil*. Other combinations of base pairings are also found in RNA, but these pairs have weak bonds between them and are much rarer and will not be considered in this work. Double helical sections are formed where canonical base pairs bond.

2.4 Function of RNA

RNA serves a diverse range of very important functions in living cells. The most important function is related to the expression of proteins from DNA encodings. Firstly mRNA (messenger RNA), which is a coding RNA, is synthesised from a gene, a DNA template, with the help of an enzyme (another type of RNA) in a process called transcription. This produces a complementary copy of the gene which encodes for a particular protein. The mRNA leaves the protected environment of the cell nucleus to carry the message to cell ribosomes. A ribosome consists of ribosomal RNA (rRNA), a non-coding or functional RNA, and protein molecules. The ribosome “reads” the mRNA message and assembles a protein according to the mRNA template.

Each triplet of nucleotides, called a codon, on the RNA encodes for 1 of 20 amino acids (see figure 2.2). The possible number of codes that can be constructed in this

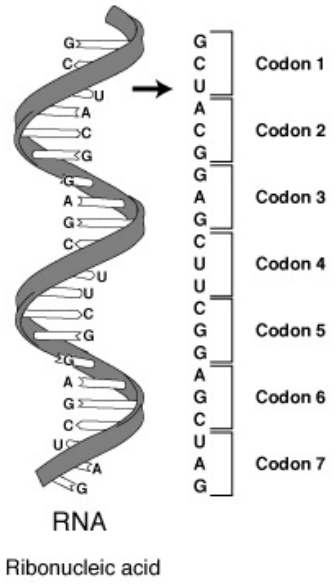


Figure 2.2: Each codon consists of three nucleotides.

way is $4^3 = 64$. The 64 codes are mapped to the 20 amino acids, so that some of the amino acids have more than one representative "code" from the 64 possible codes. This redundancy improves the reliability of the message and guards against possible fatal mutations. Some codes are reserved to indicate the start and end (start codons and end codons) of the protein string, where expression starts and stops.

Transfer RNA (tRNA), also a non-coding RNA, binds to amino acids. The tRNA molecule has an anticodon binding site (reverse of the RNA triplet or codon for that specific amino acid the tRNA is bound to) that corresponds to a codon on the mRNA. When a match is found, the amino acid binds to the previous amino acid on the chain to form a polypeptide. A reading frame is used to read the sequence, one triplet at a time. When a triplet is read, the reading frame moves forward by a full triplet. This will ensure that a single nucleotide mutation will not affect more than one amino acid.

Evidence has been found that RNA also acts as a catalyst for some biochemical reactions [44]. A review of the function of RNA as a catalyst is given in *Evolution of catalytic function* [52] and a description from this work on some of the catalytic functions follows:

5'-GGUUGCGGCCAUAUCUACCAGAAAGCACCGUUUCCCGUC
CGAUCAACUGUGUUAAGCUGGUAGAGCCUGACCGAGUAGU
GUAUGGGUGACCAUACGCGAAACUCAGGUGCUGCAAUCU-3'

Figure 2.3: 5S rRNA sequence of *Saccharomyces cerevisiae*

RNA has been shown to catalyse phosphoester transfer reactions, phosphoester hydrolysis, aminoacyl ester hydrolysis and peptide bond formation.

A review of the function of RNase P, a catalytic ribonuclease, in the maturation of tRNA and the role of self-splicing introns in the maturation of mRNA can be found in the work of Cech *et al* [21].

RNA also serves as the genetic material of some viruses, namely the retroviruses [72] [8], unlike other organisms which utilise DNA to convey their genetic information to their offspring.

2.5 RNA primary structure

Three representations of RNA structure are used. Each one represents a degree of abstraction. The primary structure is simply a one-dimensional string of characters from the set A, C, G, U representing the nucleotides in the RNA. The sequence is written, by convention, from left to right, starting at the 5' end of the string and ending at the 3' end. This representation only describes the sequence of nucleotides in the RNA strand. This is particularly useful when comparing patterns in the RNA, or doing sequence alignment to find the similarity between sequences.

Primary structure is also the easiest structure to determine in the laboratory using highly refined and efficient gene sequencing techniques. RNA sequences are derived from DNA primary sequence information using Chargaff's rules and knowledge of transcription start and ending sites. An alternative method is to generate primary sequence information from expressed sequence tags (ESTs) which provides direct information about expressed RNA in the cell. An example of an RNA primary structure is given in figure 2.3.

2.6 RNA secondary structure

The term secondary structure refers only to the base pair bindings of the RNA strand that folds onto itself and the higher order structures that emerge from this folding process. A nucleotide (base) can bind with at most one complementary base, subject to certain constraints. For this work it is assumed that only canonical base pairs can form, that is Watson-Crick pairs (AU, GC and their mirrors UA and CG) and the GU wobble pair (and its mirror UG) as these pairs are the most stable and hence the most common. The listing of these pairs is called a conformation. This is the RNA secondary structure.

Valid conformations should satisfy the following criteria [100] [101]: For any two bases $[i, j]$ and $[k, l]$ with $i < j$ and $k < l$,

1. the contacts must form canonical pairs,
2. each base must pair with only one other: $i = k$ iff $j = l$,
3. no pseudo-knots are allowed (see section 2.7), and
4. if $i < k < j$, then $i < k < l < j$.

The higher order structures mentioned previously are common RNA substructures such as hairpin loops, internal loops, bulges, multi-branch loops and dangling ends. These substructures are discussed in section 2.6.1.

Base pairs that are adjacent to each other tend to increase the stability of the conformation. These stacked pairs are commonly referred to as *stems* or *helices*. Formally, stacked pairs exist when two or more base pairs,

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

exist such that ends of the pairs are adjacent, forming a helical structure. m is used to restrict the minimum length of the nucleotide sequence connecting the helix. This connecting nucleotide sequence is the hairpin loop.

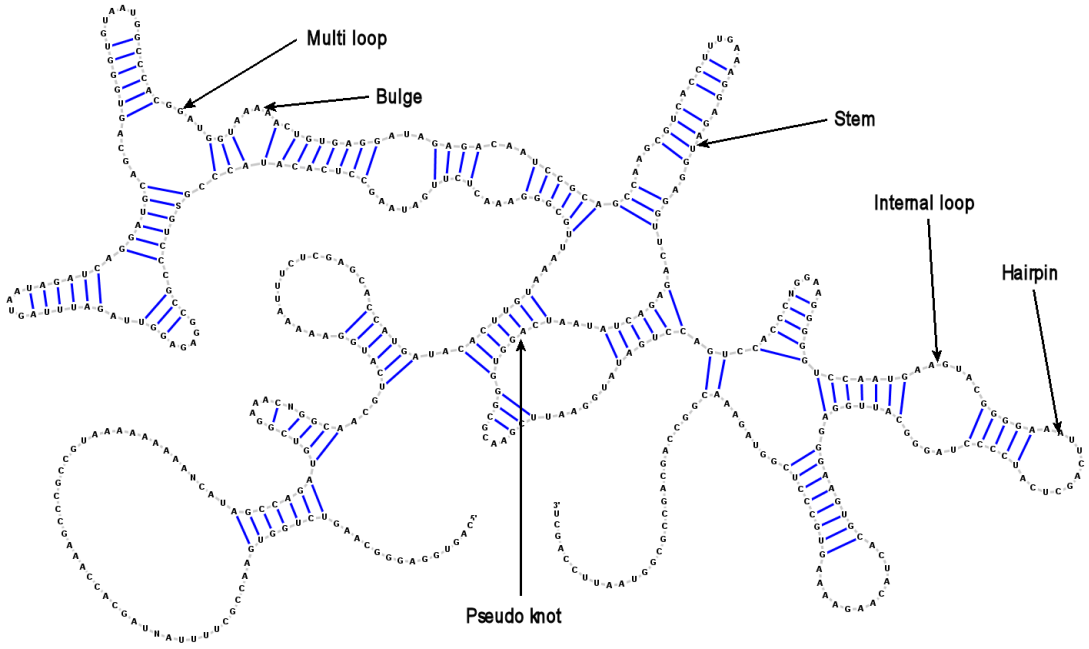


Figure 2.4: RNA secondary structure motifs. The pseudo knot is indicated but is considered part of the tertiary structure.

2.6.1 RNA secondary structure motifs

Some RNA secondary structure motifs (substructures) have been identified and described [11] [101]. These include loops such as hairpin loops, interior loops and multi-branch loops, all of which contribute to the minimization of energy of the conformation. Dangling ends, which is a string of unpaired nucleotides, can be seen at each end of the sequence. Figure 2.4 shows the most common RNA conformation substructures. These common motifs in figure 2.4 include stems, which consist of a stack of base pairs. Various loops such as hairpins, which are closed by a single stem, internal loops which are closed by 2 stems and multi loops which are closed by more than 2 stems. A special case of the loop is a bulge which shares adjacent stems on one side. The remaining motif in figure 2.4 is a pseudo knot. This is a stem that violates the properties of equation 2.1.

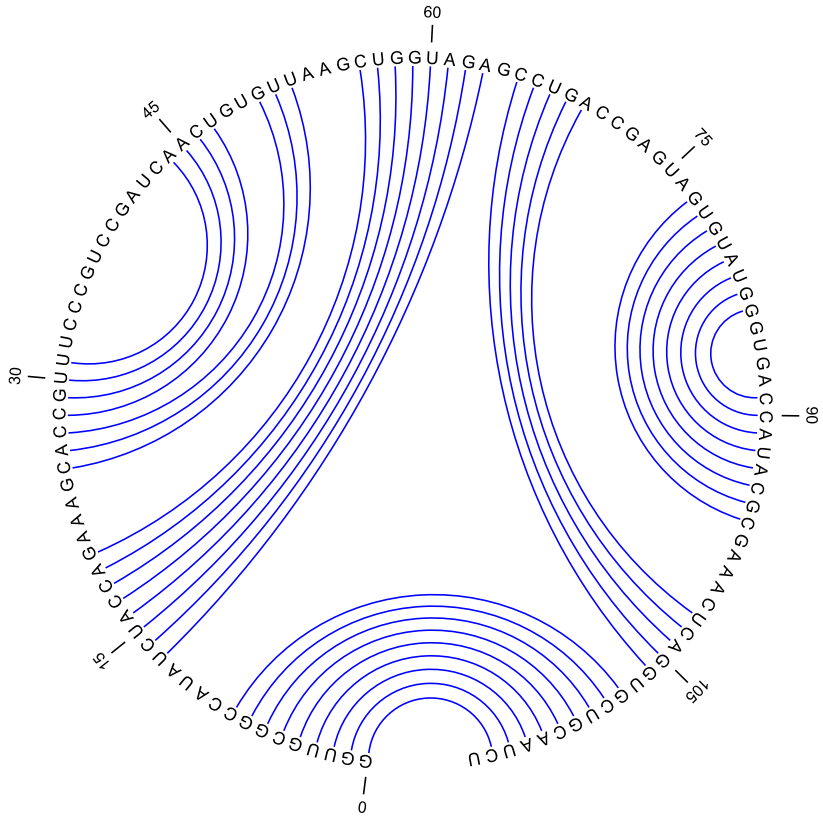


Figure 2.5: RNA secondary structure circular representation. The bands inside the circle of bases indicate which bases are paired.

2.6.2 RNA secondary structure graphical representation

The representations used in this text include graphical representations as illustrated in figure 2.4 and circular [70] representations as shown in figure 2.5. Another useful representation is the dot plot representation shown in figure 2.6. Apart from the representations used in this text, there are some other well known and useful representations of RNA secondary structure.

One representation that is easier to write is called dot-bracket notation. Dots represent bases that do not pair, and a matching opening and closing bracket marks a paired base. The dot-bracket notation is written as follows, from the 5' end of the sequence:

```
AUCAGGUUCGCAGAUUCUGGAGCUCUGAGGUCACAG
```

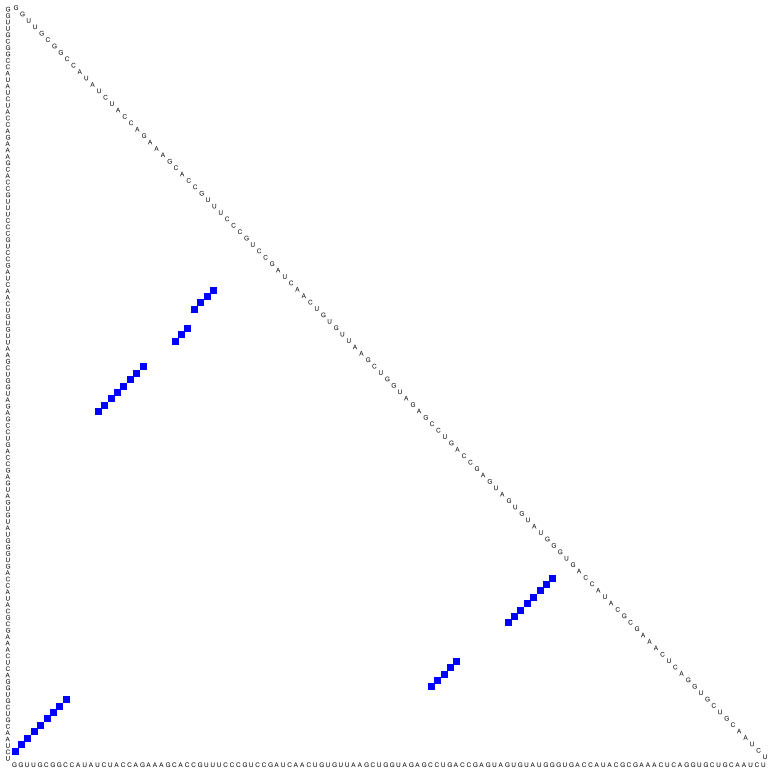


Figure 2.6: RNA secondary structure dot plot representation. The paired bases are represented by a dot in the row and column of a pair of bases.

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

A pair of brackets represents a binding between the two corresponding bases in the sequence.

2.7 RNA tertiary structure

Tertiary structure represents the RNA molecule in 3-dimensional space. Every atom has a coordinate in 3D space. Features in tertiary structure include pseudo-knots, helices, kissing hairpins and bulge contacts. Although these interactions can be represented just like secondary structure interaction (by listing of the base pairs that bind) most authors consider these interactions to be part of tertiary structure [11] [86].

2.8 Predicting the natural conformation

It is important to find the conformation or structure of biomolecules because the structure of the molecule dictates the function of the molecule. Biomolecules expose certain functional binding sites (active sites) with which to interact with other molecules. Knowing this structure, and how it interacts, would be advantageous when inventing new compounds for use in medicines that target these molecules [107] [17]. For example, the function of a targeted molecule could be inhibited when a compound is found which binds to the active site of the molecule, thereby rendering it ineffective.

Ligand docking is an example of a process that could benefit from this type of structural prediction. A small three-dimensional ligand is oriented, through an optimisation process, in such a way that it binds to an active site on a biomolecule [13].

It is thought that tertiary structure arises from the secondary structure in sequence. The thermodynamics that determine secondary structure is much more important than in the case of the tertiary structure. The base pair interactions of the secondary structure have to happen before the tertiary structural elements can form. Determining the final structure of a molecule therefore entails solving the primary, secondary and tertiary structures independently and in sequence. Finding the primary structure is relatively easy with powerful gene sequencing techniques. Determining the secondary structure is the most difficult part of the structure prediction because of the high combinatorial complexity.

2.8.1 Physical methods

There are physical methods for determining RNA secondary and tertiary structure [16] [28] [41] [65]. Unfortunately these are expensive and time-consuming experiments. A simple method of determining secondary or tertiary structure has not yet been developed.

Finding the primary structure is easy and can be done on a large scale using sequencing techniques. The primary structure provides a starting point for predicting the structure of the natural RNA conformation quickly and inexpensively using computer models.

The following subsections briefly describe the existing physical and computational

methods of determining RNA secondary structure.

X-ray crystallography

X-ray crystallography is a method used to determine the structure of molecules through X-ray diffraction [16] [28]. The closely spaced atoms in a crystal lattice diffract X-rays, which form a diffraction pattern on a recording surface. From this pattern, the position of the atoms in the crystal can be deduced. Nevertheless, there is still the problem of forming a crystal of the molecules to be studied. It is notoriously difficult to get RNA to crystallise. If crystallisation is achieved, there is still the problem that the crystallised RNA is not likely to be in a natural conformation, since the RNA conformation needs to take on a specific form in order to crystallise. Not all molecular structures are suited to crystallisation.

NMR spectroscopy

Nuclear magnetic resonance spectroscopy or NMR spectroscopy can be used to determine the physical, chemical, structural and electronic properties of a molecule. NMR can provide exact three-dimensional coordinates of the atoms in a biological molecule [41] [65].

Different atoms resonate at different frequencies when placed in the same strength magnetic field. The sample of biomolecules is irradiated with radio waves of different frequencies. The atoms absorb the energy and re-emit radio waves at a specific frequency. A receiving antenna detects these signature wavelengths.

NMR spectroscopy has been used extensively to determine the secondary structure of RNA [34] [60]. But NMR spectroscopy has its drawbacks. In the first place, expensive equipment is needed to perform NMR spectroscopy. Secondly it is possible that the RNA sample is not in the natural conformation when doing the spectroscopy, owing to environmental and temperature differences the sample has been exposed to, which hampers determination of the natural conformation.

2.8.2 Computational methods

Apart from the physical methods of determining the secondary structure of RNA, there are computational methods which are relatively inexpensive to perform. Computational methods usually employ a model of RNA folding in order to predict the structure of RNA given only the nucleotide sequence of the RNA molecule. Two computational methods are discussed next.

Free energy minimisation

Everyday prediction of RNA secondary structure has become viable with the considerable improvement in the speed and memory capacity of personal computers [90] [114] [116] [117]. The accuracy of the RNA thermodynamic models (discussed in chapter 3), used in the free energy minimisation prediction methods, have been improved with new experiments giving more accurate parameters for use in the models [62].

Free energy minimisation for RNA strands is essentially a combinatorial optimisation problem, where the objective is to find the combination of nucleotide contacts that yields the lowest free energy, ΔG.

Comparative analysis of homologous RNA sequences

Comparative analysis is more accurate than a free energy minimisation based algorithm for determining the secondary structure of RNA [108]. However, comparative analysis only works when a number of homologous RNA sequences are available. Homology is a definition of shared ancestry. Homologous sequences, which are sequences with similar origin, but found in different genomes are said to be orthologous. Homologous sequences which are found within the same genome (because of a gene duplication event), are said to be paralogous.

One of the principles in molecular biology that comparative analysis relies on is that the structure of RNA is more accurately conserved than the nucleotide sequence. Even though base pair drift occurs, the structure is conserved. For example, if one base of a pair changes, the complementary base will also change in order to keep the pair bound, and hence conserve the structure. This is called a *compensationary base change* or base

pair drift. Evidence of complementary base changes most likely indicates secondary structure base pairs.

Given a set of aligned RNA sequences, an analysis of the structures are done using a characteristic known as *mutual information content* of two columns. Columns of nucleotides appear when writing the aligned sequences underneath each other. A high correlation between two columns might indicate conserved base pairs.

To find multiple aligned sequences, a number of approaches have been developed. One of the best known is a program called CLUSTAL [43] and its descendant CLUSTAL W [92]. For an in-depth discussion on multiple sequence alignment programs and their performance, the reader is referred to [42].

A number of new approaches which combines comparative analysis (sequence alignment) and free energy minimisation have been comprehensively reviewed in [35].

2.9 Conclusion

This chapter provided an introduction to the ribonucleic acid molecule and its most important functions. Nucleotides, strands and the different structures of the RNA molecules were discussed. Primary structure is the nucleotide sequence, secondary structure is the base pairings that form, and tertiary structure is the 3-dimensional structure of the molecule. The most common RNA secondary structure motifs were also shown. Different methods of predicting the natural conformation were then highlighted. A number of physical methods and a number of computational methods were discussed. The energy minimisation method of predicting RNA conformations is of particular interest in the remainder of this work. In the remainder of this work, the RNA thermodynamic models that are used in RNA structure prediction, are described, and a new method of RNA secondary structure free energy minimisation is described and analysed.

Chapter 3

RNA Thermodynamics

The prediction of RNA secondary structure by minimising the free energy (ΔG) of RNA molecules requires a reliable thermodynamic model of RNA binding. Several thermodynamic models for RNA binding has been developed.

This chapter discusses the theory of RNA thermodynamics in section 3.1 before introducing thermodynamic models of various sophistication in section 3.2 and section 3.3.

3.1 Theory of RNA thermodynamics

Thermodynamic free energy denotes the total amount of energy in a physical system that can be converted to do work. Free energy can be expressed as a function of enthalpy, ΔH (the amount of energy possessed by a thermodynamic system to transfer between itself and the environment), entropy, ΔS (a measure of randomness or disorder of a system), and temperature T, (the measure of the average kinetic energy in a system). The free energy is calculated as follows using enthalpy, entropy and temperature:

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

Josiah Gibbs has described this equation for calculating free energy of a system as long ago as 1873 [38] [39] [40].

The free energy of each structure determines the relative amounts of each structure present at equilibrium. The Boltzmann factor, $e^{-\frac{\Delta G_S}{RT}}$, in this case, is used to compute a partition function. The partition function,

$$Q = \sum_{S \in \mathbf{S}} e^{-\frac{\Delta G_S}{RT}} \tag{3.2}$$

gives a Boltzmann weighted counting of all structures with different free energies, ΔG_S, at a certain temperature [66], where S is a structure, R is the gas constant and T is the absolute temperature of the system. According to statistical mechanical theory, this weighting gives the probability density for every folding S. That is, the probability of any particular folding, S, is given by

$$\frac{e^{\frac{-\Delta G_S}{RT}}}{Q} \tag{3.3}$$

The concentration ratio

$$\frac{[C_1]}{[C_2]} = e^{-\frac{\Delta G}{RT}} \tag{3.4}$$

where C_1 and C_2 represent the concentrations of two different structures in equilibrium in a system. The concentration of structures varies exponentially with free energy, ΔG, as seen in equation (3.4). Small differences in free energy translate into relatively large differences between concentrations of specific structures. Because the RNA folding energy surface is not smooth, it contains many local optimum conformations. Changing the conformation by a few base pairs greatly impacts the free energy of the conformation and hence the conformation's relative concentration in the system.

The secondary structure motifs discussed in section 2.6.1 and presented in figure 2.4 contribute the most to the free energy calculation. A motif's contribution can be calculated using thermodynamic parameters determined for the type of motif. Each motif is independent of the other in the way the thermodynamic parameters are described, giving them additive property, meaning the contributions of each motif (i.e. helices, loops and hairpins) are simply added together to obtain the total free energy of the conformation [7].

Relatively good thermodynamic models are needed to produce accurate predictions of RNA secondary structures. Although much research has been done on refining these

thermodynamic models over the years, they are not perfect yet [116]. Some models are too simple and do not capture all the elements that contribute to free energy. The thermodynamic energy parameters are also derived from noisy data. Further, there are no parameters to describe some of the secondary structure motifs that are present in actual RNA molecules. This uncertainty in the thermodynamic models translates to uncertainty in the free energy predictions of conformations. It is believed that the native conformation of RNA molecules exist close to or at the minimum free energy state. It is also possible that the RNA molecules exist as a distribution of low free energy structures which may fluctuate between the different states [50] [51] [63].

3.2 Hydrogen bond models

The hydrogen bond model for RNA thermodynamics is primarily used by an RNA secondary prediction algorithm called RnaPredict [106]. Hydrogen bond models are extremely simple to implement and quick to evaluate due to the simple energy rules it depends on.

The hydrogen bond model assigns a free energy change to the formation of a base pair. This means that an energy value is assigned to each possible canonical base pair. When this pair appears in a structure, the related energy value is added to the structure's total energy.

3.2.1 Major model

Wiese and Glen proposed using a simple scheme devised by Major, where the free energy change introduced by the forming of base pairs, are summed [106]. The following values are used in formation of base pairs:

$$\Delta G(GC) = \Delta G(CG) = -3kcal/mol$$
$$\Delta G(AU) = \Delta G(UA) = -2kcal/mol$$
$$\Delta G(GU) = \Delta G(UG) = -1kcal/mol$$

To calculate the free energy of a given structure, the sum of energies of all pairs is

computed:

$$\Delta G(S) = \sum_{i,j \in S} e(r_i, r_j)$$

where $e(r_i, r_j)$ is the free energy between the ith and the jth nucleotide of the structure that forms a base pair. This is a very simple thermodynamic energy model which accounts for the bonding energies between the base pairs only. The GC pair has three hydrogen bonds, the AU pair has two hydrogen bonds, and the wobble pair GU has a weaker bonding than the AU pair. Thus the UG pair contribute only $-1kcal/mol$ to the total energy, although it has 2 hydrogen bonds.

3.2.2 The Mathews model

The Mathews model is based on the same principal of adding up the free energy contribution of the individual base pair combinations [106]. Instead of using the approximate proportional stability of the base pairs, the actual number of hydrogen bonds is used:

$$\Delta G(GC) = \Delta G(CG) = -3kcal/mol$$
$$\Delta G(AU) = \Delta G(UA) = -2kcal/mol$$
$$\Delta G(GU) = \Delta G(UG) = -2kcal/mol$$

The GC pair has three hydrogen bonds, the AU pair has two hydrogen bonds, and the GU pair has two hydrogen bonds, hence the UG pair contribute $-2kcal/mol$ to the total energy. Again, to calculate the structure's free energy, the free energy contributions from each base pair are summed [62].

3.2.3 Limitations of the hydrogen bond model

The hydrogen bond model assumes that the identity of the individual base pairs determines the free energy contribution. This is a reasonable approximation because each base pair bond reduces the amount of free energy of the structure. But this model fails to include the intramolecular contributions, such as stacking energies and loop strain, that exist in regular RNA molecules [106]. A more sophisticated energy model was therefore developed which does take into account stacking energies and loop contributions. This energy model is described in the next section.

3.3 Stacking energy model

Stacking energy models used in more advanced algorithms, e.g. *mfold* [115] and Vienna RNA package [45] contain two important components: reliable energy parameters and a statistical mechanical model. The statistical model and energy parameters are determined in the laboratory [33] [58]. This process basically entails synthesising the RNA sequence of interest. When a large enough quantity of the RNA is synthesised, it is subject to radiation. Chemical substances absorb radiation at specific frequencies. The amount of radiation absorbed for a specific frequency is related to the concentration of the chemical substance. By changing the temperature and monitoring the absorbency of radiation using a spectrophotometer, the curve of absorbence versus temperature can be plotted. From this plot, the energy needed to break or melt the base pairs can be determined. The free energy loss (ΔG) for the formation of the synthesised base pair can now be determined from this melting energy [78].

By determining melting energies for all possible combinations of base pairs, and solving sets of equations, thermodynamic parameters for adjacent pairs can be calculated [62].

For a number of years the Turner laboratory has been estimating the nearest neighbour parameters for synthetically constructed RNA oligoribonucleotides [32] [50] [81] [90] [91] [99] [109].

Two basic stacking energy thermodynamic models exist: individual nearest neighbour (INN) (Serra and Turner, [81]) and individual nearest neighbour-hydrogen bond (INN-HB) (Xia *et al.*, [110]). The essential idea behind the INN and INN-HB stacking-energy models is that the stabilizing contribution each base pair makes to its helix depends on that base pair's nearest neighbours. For example, the free energy contribution of a GC base pair would vary depending on whether the adjacent base pair in the helix is either an AU base pair, or its mirror a UA base pair.

3.3.1 Individual nearest neighbour model

There are two distinct components to computing the free energy of a helix using INN. The first is initiation, or the formation of the first base pair. Initiation brings the two

strands together and entails hydrogen bonding. The second component is propagation, or the continued formation of subsequent base pairs. Propagation involves nearest-neighbour or stacking interactions as well as hydrogen bonding. The nearest-neighbour thermodynamic parameters used in the INN model were initially measured at 25 ℃, [7], but were later remeasured and extended at 37 ℃. A thorough review of the INN model complete with thermodynamic parameters can be found in [81].

3.3.2 Individual nearest neighbour-hydrogen bond model

Later experimentation determined that duplexes with identical nearest neighbours but varying terminal ends also differed in their stabilities. Specifically, a duplex with one additional terminal GC pair and one less terminal AU pair is always more stable [110]. This new thermodynamic detail is added to the INN model, resulting in the individual nearest-neighbour hydrogen bond model. Although the INN-HB model only specifies a penalty for terminal AU pairs, the terminal GU pairs are given the same penalty as suggested by Mathews *et al.* [62].

3.4 Conclusion

Chapter 3 discussed different thermodynamic models for RNA conformation. The simplistic hydrogen bond models were discussed first. These models do not capture all the effects of the loops and the destabilising effects of other substructures. A more sophisticated thermodynamic model, the stacking-energy model was then discussed. The stacking energy model is used in the remainder of this work to make predictions of RNA conformations.

Chapter 4

Particle Swarm Optimisation

Nature has inspired many algorithms in computer science applications, for example artificial neural networks for pattern recognition or genetic algorithms and ant colony simulations for solving optimisation problems. Nature has solved a lot of hard problems already, and by closely observing Nature, one can deduce the principles of its efficient processes and apply those principles to computational problems. Particle swarm optimisation is another algorithm inspired by Nature. In this chapter, the PSO algorithm is introduced in section 4.1. Section 4.2 describes the basic PSO algorithm while the following sections, sections 4.3 and 4.4 describe the swarm structures and parameters used in the PSO algorithm.

4.1 Introduction

The particle swarm optimisation (PSO) algorithm is a stochastic optimisation algorithm developed by Kennedy and Eberhart in 1995 [55], and has subsequently proved to be a better optimisation algorithm than evolutionary computation (EC) algorithms including genetic algorithms [22] [24] [25] on certain complex computational problems. PSO is an elegant and simple algorithm modelled after the flocking behaviour of birds. It was initially intended to be a visualisation of the unpredictable choreography of a flock of birds in flight. From these visualisation experiments, a very efficient optimisation algorithm was developed.

Individuals or "particles" in a swarm, fly around in a multidimensional search space. The movements of these particles are influenced by their own successes and the success of other particles in the swarm. Each particle shares information about its own best solutions thus far so the swarm knows the most promising areas in the search space. Particles in the swarm tend to return to previous successful regions in the search space.

A social structure exists within the swarm. This social structure determines the communication channels that exist between individuals. This structure is in the form of a neighbourhood. A number of neighbourhood topologies have been invented for use by the PSO like the star, ring, wheel and Von Neumann network topologies. These social structures are discussed in section 4.3. The basic particle swarm optimisation algorithm is described next.

4.2 PSO algorithm

This section describes the basic PSO algorithm as invented by Kennedy and Eberhart in 1995. The algorithm, swarm social structures and algorithm parameters are discussed in the following subsections.

4.2.1 Position and velocity

The PSO algorithm works by moving the particles through the search space. A particle represents a potential solution in the search space. Each particle has a position vector, $\mathbf{x}_i$, and a velocity vector $\mathbf{v}_i$. This position is updated at every iteration by adding the particle's current velocity, $\mathbf{v}_i$, to the position vector. $\mathbf{x}_i(t)$ denotes the position of the particle at time-step t, where t is a discreet time step. The position update is done via the update equation, as follows

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{4.1}$$

with $\mathbf{x}_i(0) \sim U(\mathbf{x}_{min}, \mathbf{x}_{max})$.

The velocity vector is updated at every iteration with social and cognitive components. The social component reflects the social knowledge about the best solution found

so far in the particle's neighbourhood. When the best particle in a specified particle's neighbourhood is used to provide social knowledge, the PSO algorithm is called local best PSO or *lbest* PSO for short. A special case exist where the neighbourhood encompasses the whole swarm and is called global best PSO or *gbest* PSO. The cognitive component reflects the particle's own experience and represents the best solution the particle itself has found so far. Velocity is a linear combination of the previous velocity, the social component, and the cognitive component and is updated as follows:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)] \tag{4.2}$$

where $v_{ij}(t)$ is the velocity of particle i in dimension $j = 1, ..., n_x$ at time step t, $x_{ij}(t)$ is the position of particle i in dimension j at time step t, c_1 and c_2 are positive acceleration constants used to scale the contribution of the cognitive and social components respectively (discussed in section 4.4.2), and $r_{1j}(t), r_{2j}(t) \sim U(0,1)$ are random values in the range [0,1], sampled from a uniform distribution. These random values introduce a stochastic element to the algorithm.

The personal best position, $\mathbf{y}_i$, associated with particle i is the best position the particle has found so far, which is updated as follows:

$$\mathbf{y}_i(t+1) = \begin{cases} \mathbf{y}_i(t) & \text{if } f(\mathbf{x}_i(t+1)) \geq f(\mathbf{y}_i(t)) \\ \mathbf{x}_i(t+1) & \text{if } f(\mathbf{x}_i(t+1)) < f(\mathbf{y}_i(t)) \end{cases} \tag{4.3}$$

where f is the fitness function to be minimised.

The global best position, $\hat{\mathbf{y}}_j(t)$, at time step t can be calculated as follows:

$$\hat{\mathbf{y}}_j(t) \in \{\mathbf{y}_0(t), ..., \mathbf{y}_{n_s}(t)\} \,|f(\hat{\mathbf{y}}(t)) = min\,\{f(\mathbf{y}_0(t)), ..., f(\mathbf{y}_{n_s}(t))\} \tag{4.4}$$

where n_s is the total number of particles in the swarm.

A variation on calculating the global best is to select the particle with the best fitness in the *current* iteration as the best solution. Equation (4.4) then becomes

$$\hat{\mathbf{y}}_j(t) \in \{\mathbf{x}_0(t), ..., \mathbf{x}_{n_s}(t)\} \,|f(\hat{\mathbf{y}}(t)) = min\,\{f(\mathbf{x}_0(t)), ..., f(\mathbf{x}_{n_s}(t))\} \tag{4.5}$$

where $\mathbf{x}_i(t)$ is the current position of particle i.

4.2.2 Algorithm

A concise pseudo code PSO algorithm is presented in algorithm 4.1.

First, a swarm S, of size n_s, particles is initialised to random positions in the search space. The algorithm then begins to iterate and performs the following steps:

- The personal best position of each particle is determined, after which the global best particle of the swarm is determined.
- The velocity of each particle and its resulting new position is calculated.

The algorithm keeps on performing these iterations until a stopping condition, such as swarm convergence or maximum number of iterations, is met.

4.3 Swarm structure

The driving force behind PSO is the social interaction between particles and their own memory of good solutions. Particles will try to emulate their "better" neighbours and move towards these neighbours in the neighbourhood and also to their own best experiences. These particle neighbourhoods are defined in terms of particle labels and not in terms of topological information like Euclidean distance measures [53].

The star topology [53], also known as *gbest* topology, arises when all the particles are in the same neighbourhood, that is, a fully interactive social network exists where each particle can communicate with all other particles. The entire swarm is then attracted to the single best particle in the swarm.

In the ring topology [53], each particle only communicates with its n immediate neighbours. Each particle attempts to move closer to the best particle in its neighbourhood. The *lbest* version of PSO uses this topology. The *gbest* PSO is a special case of the *lbest* PSO where the entire swarm is the neighbourhood.

In the wheels topology [53], all the particles communicate with only one specific particle. The individuals are effectively isolated from each other. The middle particle

Create and initialise a swarm, S, of size n_s
repeat
  for *each particle* $i = 1, ..., S.n_s$ **do**
    // set personal best position
    if $f(S.x_i) < f(S.y_i)$ **then**
      $S.y_i = S.x_i$
    end
    // set global best position
    if $f(S.y_i) < f(S.\widehat{y})$ **then**
      $S.\widehat{y} = S.y_i$
    end
  end
  for *each particle* $i = 1, ..., S.n_s$ **do**
    update the velocity using equation (4.2)
    update the position using equation (4.1)
  end
until *stopping condition is true*

Algorithm 4.1: Pseudocode outline of the *gbest* PSO algorithm

moves in the direction of the best particle. If this middle particle's position improves, all other particles will then be attracted to its position.

The Von Neumann topology arranges the particle's social structure in a grid structure. This topology has been shown to outperform other social network structures on a large number of problems in a number of empirical studies [24] [56] [74]. The Von Neumann topology strikes a good balance between a fully connected swarm which converges quickly and a loosely connected swarm which covers a larger search space.

4.4 PSO control parameters

The PSO algorithm is very sensitive to its control parameters. In this section, the PSO inertia weight, acceleration constants cognitive and social components and a maximum velocity constraint are discussed.

4.4.1 Inertia weight

Performance of the PSO algorithm can be improved by introducing an inertia weight ω, [88], that is used to control the influence of the previous velocity on the current velocity:

$$v(t) = \omega v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)] \tag{4.6}$$

Larger inertia weights cause more exploration of the search space while small inertia weights focus the search on a smaller region. The inertia weight in equation (4.6) is important for convergent behaviour [87]. When $\omega \geq 1$, velocities will increase over time, and the swarm will diverge. For $\omega < 1$, particles decelerate, depending on the values of c_1 and c_2, until their velocities reach zero. Larger values of ω enhance the explorative behaviour of the swarm while smaller values of ω promotes local exploitation. The choices of parameter values for ω and c_1 and c_2 are important to ensure convergent particle trajectories. Various studies have been done on parameter values and suggestions have been made about the parameter value combinations to be used for ω, c_1 and c_2, discussed in great detail in [24].

4.4.2 Cognitive and social components

The **cognitive component**, $c_1\mathbf{r}_1(\mathbf{y}_i - \mathbf{x}_i)$, represents the particle's performance relative to its own experience. The effect of this component is that the particle tends to return to the position where it experienced the best fitness value. Kennedy and Eberhart referred to this component as the "nostalgia" of the particle [55].

The **social component**, $c_2\mathbf{r}_2(\hat{\mathbf{y}} - \mathbf{x}_i)$ in the case of *gbest* PSO or $c_2\mathbf{r}_2(\hat{\mathbf{y}}_i - \mathbf{x}_i)$ in the case of *lbest* PSO, represents the performance of the particle relative to the best solution found so far in the neighbourhood. The social component will tend to direct the particle

to the neighbourhood best solution.

4.4.3 Acceleration constants

As mentioned previously, the acceleration constants determine the amount of influence from the *social* or *cognitive* components in the velocity update. A comparatively larger c_1 will bias the movement towards the particle's own best position whereas a comparatively larger c_2 will bias the movement towards the best particle in the neighbourhood. This means that when c_1 is large, particles tend to explore more and conversely, when c_2 is comparatively large, the swarm is attracted to a single point $\hat{\mathbf{y}}$. If $c_1 = c_2$, particles are attracted towards the average of $\mathbf{y}_i$ and $\hat{\mathbf{y}}$. Some heuristics for choosing values for c_1 and c_2 have been derived theoretically by Trelea [94] and Van den Bergh and Engelbrecht [98].

Trelea suggested choosing values under the constraint [94]

$$\omega > \frac{1}{2}(c_1 + c_2) - 1 \tag{4.7}$$

while Van den Berg and Engelbrecht proved that for an unconstrained simplified PSO with inertia, the particle trajectory converges if the following conditions hold [98]

$$1 > \omega > \frac{1}{2}(c_1 + c_2) - 1 \geq 0 \tag{4.8}$$

4.4.4 Maximum velocity

A maximum velocity could be imposed on the particles to limit step sizes and thus limit the particles from accelerating out of control and causing divergent particle trajectories. Using maximum velocity restrictions is however problem dependent and is not always needed.

One solution to impose a maximum velocity is by velocity clamping [23] [89] where a particle's velocity is set to the maximum velocity whenever the particle's velocity exceeds that maximum velocity:

$$v_{ij}(t+1) = \begin{cases} v'_{ij}(t+1) & \text{if } v'_{ij}(t+1) < V_{max,j} \\ V_{max,j} & \text{if } v'_{ij}(t+1) \geq V_{max,j} \end{cases} \tag{4.9}$$

where v'_{ij} is calculated using equation (4.2) or (4.6).

Velocity clamping is a very simplistic way of trying to minimise swarm divergence, provided the correct value of V_{max} is used. Unfortunately, velocity clamping when done in this manner has a few unwanted side-effects. Firstly, velocity clamping not only changes the step size, but can also change the direction the particle moves in. Consider when one of the components of $\mathbf{v}'_i$ say v'_{ij} is above the $V_{max,j}$ threshold, only that particular component, v'_{ij}, is changed, which results in a change of the direction of vector $\mathbf{v}'_i$. Secondly, a problem with velocity clamping occurs when all velocities are equal to the maximum velocity. Particles will remain on the boundary of a hypercube defined by $[\mathbf{x}_i(t) - \mathbf{V}_{max}, \mathbf{x}_i(t) + \mathbf{V}_{max}]$. This problem can be resolved by introducing a inertia weight (refer to section 4.4.1). Alternative velocity clamping solutions can be found in [24].

4.5 Conclusion

This chapter introduced the particle swarm optimisation algorithm developed by Kennedy and Eberhart in 1995. The emergent swarm behaviour of the particles in the swarm has proved to be a simple but very reliable optimisation technique. After a general discussion of the PSO algorithm, a pseudo code algorithm was given. Subsequently, the different parameters of the algorithm were discussed in detail. The different topologies for swarm communication were also highlighted. In the following chapter, the PSO algorithm is adapted to a discrete-valued problem domain, the domain of mathematical sets.

Chapter 5

Set Particle Swarm Optimisation

This chapter proposes a new version of the PSO algorithm [55] described in the previous chapter, namely SetPSO which operates on mathematical sets in order to solve set-based combinatorial problems. The particle representation in SetPSO is introduced before the new operators of the SetPSO algorithm is introduced. Finally, the complexity of the new SetPSO algorithm is analysed.

5.1 Introduction

The original PSO algorithm was designed to work in continuous-valued search space. Not all optimisation problems are set in the continuous-valued domain though. SetPSO therefore was designed to work with discrete-valued search spaces, in an attempt to extend the powerful optimisation capabilities of PSO to the discrete-valued domain. More specifically, SetPSO [68] is the first PSO adapted to work on mathematical sets which can contain a variable number of elements. In order to adapt PSO, new operators are defined to work on sets instead of the usual continuous vector space.

SetPSO is not the first variant of PSO which changes the particle representation and the meaning of the position and velocity vectors of the original PSO in order to adapt PSO to new domains. Binary PSO, created by Eberhart and Kennedy [54] changes the representation of the position vector in Binary PSO into a bit-string, while the velocity is used to calculate a probability of a bit being flipped in the binary position string.

Clerc [15] and Schoofs and Naudts [79] also modified particle swarm optimisers to solve discrete-valued problems by defining new abstract arithmetic operators.

The solution space for the SetPSO and the representation of a particle's position are discussed in section 5.2.1. The redefinition of the addition and subtraction operators is discussed in sections 5.2.2 and 5.2.3 respectively. The distance operator, which is used in a swarm diversity measurement later in this work, is defined in section 5.2.4. A pseudo code algorithm of the proposed algorithm is discussed in 5.3. Particle initialisation is discussed in section 5.3.1 and the new velocity update and position update operations are shown in sections 5.3.2 and 5.3.3. Finally, the new parameters for SetPSO are defined and discussed in section 5.4.

5.2 Defining SetPSO

This section presents the necessary particle representation, operator definitions and velocity and position update operators used in SetPSO.

5.2.1 Solution space and particle position

The solutions (particle positions), S_i, generated with the SetPSO algorithm are mathematical sets. Therefore, elements within a solution are unique. The solution to a problem is a subset $S \subseteq U$, which is a combination of elements from the *universal set* U. The universal set can be either finite or infinite, although the solution sets S_i are always finite.

It is possible that the problem to be solved imposes some constraints on the solution sets. For example, some elements in the universal set are not compatible with each other and cannot exist together in a solution set. It is important that the implementation of the operators modify the solution sets in such a manner that the solution sets are always valid under the constraints set by the problem. This is certainly the case in this SetPSO implementation to solve RNA secondary structures, as shown in chapter 8.

In order to change the vector space from the basic PSO into the set space the SetPSO algorithm operates in, only three operators need to be redefined, namely the

- addition operator, the
- subtraction operator, and the
- distance operator.

The addition operator and subtraction operator have their naive set theory definitions. Subsequently, it is shown how each of these operators is applied in the new SetPSO algorithm. The multiplication operator retains its normal mathematical function and is only used to multiply scalar values in SetPSO.

5.2.2 Addition operator

The addition operator "adds" two sets together. The operator is also known as the union operator in set theory. Given two sets, $\mathbf{A}$ and $\mathbf{B}$, their union is written as

$$\mathbf{A} \cup \mathbf{B}$$

or

$$\mathbf{A}+\mathbf{B}$$

5.2.3 Subtraction operator

The subtraction operator, also known as the *relative complement* or *set theoretic difference* between two sets $\mathbf{A}$ and $\mathbf{B}$, is written as

$$\mathbf{A}\text{-}\mathbf{B}$$

or

$$\mathbf{A} \setminus \mathbf{B}$$

and denotes the set of all elements which are members of $\mathbf{A}$ but not members of $\mathbf{B}$.

5.2.4 Distance operator

The metric that computes the distance between two sets is based on the well-known *string edit distance* metric. The Levenshtein distance [61] is a string edit metric, given by the minimum number of operations needed to transform one string into another, where an operation is an insertion, deletion or substitution of a character. However, it is no trivial task to compute the *minimum* number of operations needed to transform one string into another using the Levenshtein metric. Usually a dynamic programming algorithm is used to compute the edit distance. For the purpose of measuring the distance between two sets, the substitution operation is not considered.

The distance metric for sets is given by the number of operations needed to transform one set into the other using insertion (adding an element to a set) and deletion (removing an element from a set) operations only. Given a subject set $\mathbf{A}$, and a target set $\mathbf{B}$, the number of operations needed to do the transformation from $\mathbf{A}$ to $\mathbf{B}$ can be calculated using the previously defined set subtraction operator in conjunction with the set size operator. First, the size of the set of elements present in $\mathbf{A}$ and not present in $\mathbf{B}$ is calculated. This represents the number of deletion operations necessary. Then the size of the set of elements present in $\mathbf{B}$ but not in $\mathbf{A}$ is calculated. This represents the number of insertion operations required. The sum of these two calculations gives the edit distance. In set notation the distance between $\mathbf{A}$ and $\mathbf{B}$ is:

$$|\mathbf{A} - \mathbf{B}| + |\mathbf{B} - \mathbf{A}|$$

Remember that, "+" has its normal arithmetic meaning here as the result of a size operation is a scalar value.

5.3 SetPSO Algorithm

The algorithm differs from the basic PSO in the velocity update and position update steps. Algorithm 5.1 outlines the new algorithm. In this algorithm, the notation $S.X_i$ is used to denote the position X of particle i in the swarm S. $S.Y_i$ is the personal best position of particle i while $S.\widehat{Y}_i$ represent the neighbourhood best position.

Create and initialise a swarm, S, of size n_s (see section 5.3.1)
repeat
  for *each particle* $i = 1, ..., S.n_s$ **do**
    // set personal best position
    if $f(S.X_i) < f(S.Y_i)$ **then**
      $S.Y_i = S.X_i$
    end
    // set neighbourhood best position
    if $f(S.Y_i) < f(S.\widehat{Y})$ **then**
      $S.\widehat{Y} = S.Y_i$
    end
  end
  for *each particle* $i = 1, ..., S.n_s$ **do**
    update the velocity (see section 5.3.2)
    update the position (see section 5.3.3)
  end
until *stopping condition*

Algorithm 5.1: Pseudo code outline of the SetPSO algorithm.

5.3.1 Particle initialisation

Each particle is initialised to a random subset of the (finite or infinite) *universal set* U; that is,

$$S.X_i \subseteq U,\ |S.X_i| \leq |U|$$

and all elements of $S.X_i$ are unique.

If there are constraints specified on the solution sets, the particles should be initialised in a manner that does not violate any constraints. In this case, particles are initialised with stems that do not contain conflicting base pairs.

5.3.2 Velocity update

The velocity of a particle is actually represented by two sets of elements: The first set contains the elements that should be **removed** from the current position set. This first set is called the *open set* or O. Removing the *open set* from the current position produces an intermediate set, because the resulting set is not yet a particle position. The second set contains the elements that should be **added** to the intermediate set. This second set is called the *close set* or C.

Calculating the *open set*

The *open set* is a random subset of the current position. The elements in O will be removed from the current position in the position update step. The *open set* is computed as follows:

for each element e in X
  $r \sim U(0,1)$
  if $r < P_I$
    add e to O
end for

where P_I is an entropy weight. The entropy weight is discussed in section 5.4. The entropy determines how many of the elements in the current position are removed.

Calculating the *close set*

The *close set* is a random combination of elements from the target set B and the universal set U.

Step 1

Add the *pbest* and *lbest* positions sets together to construct the target set, i.e.

$$B = S.Y \cup S.\widehat{Y} \tag{5.1}$$

The target set, B is used to direct the particle towards its own personal best position and the neighbourhood best position similar to the velocity update of the original PSO.

Step 2

for each element e in B
  $r \sim U(0,1)$
  if $r < P_C$ and e not in X
    add e to C
end for

where P_C is a *closing probability* weight used to control the influence of the target set. P_C is discussed in section 5.4.

Step 3

for each element e in U
  $r \sim U(0,1)$
  if $r < P_R$
    add e to C
end for

where P_R is a *random add* probability weight and determines the rate at which new elements are introduced into the particle's position set. P_R is discussed in section 5.4.

The elements included in the *close set* are added to the current position of the particle in the position update step, as explained in section 5.3.3. This tends to move the current particle towards the target position. In order to introduce diversity, random elements from the *universal set* are added to the *close set* in step 3 of the velocity update. If this is not done, the particles would explore only combinations of the elements with which the swarm was initialised.

In some combinatorial optimisation problems, like this implementation of RNA secondary structure prediction, all the elements in U might not be compatible with each other and cannot exist in the same solution set. Refer to chapter 7 for a discussion of

the constraints on elements of this RNA structure prediction method. The constraints on the compatibility of the elements need not be considered in the velocity update. Incompatible stems are allowed to be in the *close set* as the constraints are enforced in the position update step, described in the next section. This saves computation time during the velocity update step.

5.3.3 Position update

The objective of the position update is to create a new candidate solution. In this process the current position set, $S.X_i$, for particle i is modified using the particle's velocity to produce a new position $S.X_i''$. The position update occurs in two steps.

The first step removes all the elements in the *open set* O from the current position $S.X_i$ to give an intermediate set $S.X_i'$:

Step 1

$$S.X_i' = S.X_i - O \tag{5.2}$$

Remember that the *open set* O contains elements that are in the current position $S.X_i$. Elements are removed from the current position to make space for new elements introduced to the particle position in step 2.

The second step adds all the elements in the *close set* C that will not cause the new solution set to violate any constraints, if any, to $S.X_i'$. $S.X_i''$ represents the new particle position.

Step 2

$$S.X_i'' = S.X_i' \cup C \tag{5.3}$$

Step 2 introduces the new elements that might be contained in C to the particle's position.

The basic position update procedure would just add all of the elements of C to the particle position in step 2, except of course where it would introduce a duplicate element into the position set (sets contain unique elements by definition). Where constraints on the solution sets are specified, for example when some elements are incompatible in a

solution set, a specialised addition operator is implemented. The specialised addition operator verifies that the resulting set is valid under the constraints specified by the problem.

Further modification to the specialised addition operator can be made. The addition of elements to a set under constraints can be seen as an optimisation process in itself. A simple and naive optimisation technique is to give the addition operator hill climbing characteristics. A hill climbing addition operator then adds the element in C to the $S.X_i'$ set in a greedy fashion, i.e. the element which increases the fitness of $S.X_i'$ the most is added first. Then the next best element is added, and so forth. The hill climbing addition operator forces the particle to the best position it can be in under the constraints and leads to quicker convergence. Unfortunately, this approach has the drawback that it requires a large number of comparisons and thus is considerably slower than a non-optimising addition operator. Of course, other optimisation techniques may be used in the implementation of the optimised addition operator. Refer to section 7.4 for a discussion on the optimised addition operator used in this dissertation.

5.4 SetPSO parameters

The performance of the SetPSO is influenced by three parameters, namely the *closing probability*, P_C, the *random add probability*, P_R, and the *entropy* weight, P_I. These parameters are described in detail in this section.

5.4.1 Closing probability

The *closing probability* parameter, P_C, is analogous to the social and cognitive component parameters in the original PSO. P_C controls the way both the neighbouring particles and the particle's own memory influence the position of the particle. P_C determines the probability that an element contained in the particle's personal best position or in the neighbourhood best position will be added to the *close set* of the particle's velocity (see step 2 of calculating the *close set* in the previous section). Theoretically, the greater the value of P_C, the more influence the particle's own best position and the neighbourhood best position will have in the particle position update. This should speed up conver-

gence of the swarm. Conversely, the lower the value of P_C, the less the influence of the previously found "good" solutions will be in the particle position update step of SetPSO.

5.4.2 Random add probability

The *random add probability*, P_R, controls the probability that new random stems will be added to the *close set* of the particle's velocity. Thus, P_R determines the amount of diversity introduced to the solutions. The higher the value of P_R, the better the chance that a randomly selected stem from the *universal set*, U, will be added to the *close set* C. The consequence of having a small value for P_R is less diversity in the swarm, which leads to less exploration of the search space. In the border case where $P_R = 0$, no new elements from U can be added to a particle's position apart from the elements contained in the neighbourhood best position. In the case of $P_R = 0$, all the particles' positions will be a subset of the elements with which the swarm is initialised.

5.4.3 Entropy weight

The *entropy* weight parameter, P_I, is analogous to the *inertia* weight in the original PSO. The reason that P_I is analogous to the inertia weight is because it determines what influence the particle's current position has on the future position of the particle. This parameter controls the probability of stems being added to the *open set* of the velocity. Hence P_I controls the size of the *open set* and the "disruption" that the removal of elements from the current solution set will cause in the position update (see step 1 of the position update operation). The greater the entropy probability, the greater the disruption to the solutions. Disruption simply means the particle loses many of its elements which can be replaced by new elements, improving swarm diversity. The consequence of a low *entropy* is that the swarm change less and less randomness is introduced into the swarm.

5.5 Diversity measurement

Diversity measurements provide an indication of how diverse the population of solutions are, or stated differently, how dissimilar the solutions are. By measuring the diversity of the swarm when using varying values for the P_I, P_C and P_R parameters, the combination of P_I, P_C and P_R result in the most diverse swarms, or which one of the three parameters has the most significant impact on the diversity of the swarm, can be determined.

A simple swarm diameter measurement determines the diameter of a swarm of particles in the classic PSO, by calculating the Euclidean distance between the particles. Using swarm diameter as a diversity measurement in SetPSO implies measuring distance between particles that consists of sets. Therefore, a distance operator for sets has been proposed in section 5.2.4. The diversity of the swarm is determined by calculating the average distance, $\mathbf{D}$, of the particles in the swarm to the *gbest* particle [76]. Obviously, the larger the average distance from the *gbest* particle (or swarm diameter), the more diverse the swarm is, as the particles cover more of the search space. Diversity is calculated by the following equation,

$$\mathbf{D}(S(t)) = \frac{\sum_{i=1}^{n_s} \left(\left| S.X_i(t) - S.\widehat{Y}(t) \right| + \left| S.\widehat{Y}(t) - S.X_i(t) \right| \right)}{n_s} \tag{5.4}$$

where n_s is the size of the swarm, $S.X_i$ is the current position of particle i and $S.\widehat{Y}$ is the position of the *gbest* particle.

The swarm diameter measurement can be taken after each iteration of the SetPSO algorithm and is simply used to track the diversity of the swarm during an experiment.

5.6 Computational complexity

Computational complexity provides an indication of the extent of the resources needed to perform a computation. These resources are time and memory. Usually there is a trade off in computations between time and complexity. The more complex a computation is, the more time it will take to execute. Time complexity is estimated as the number of steps for a computation to complete while space complexity is estimated as the number of bits the computation needs to store data while the computation executes. Complexity

is usually expressed as a function of some input parameter.

The time complexity for *mfold*, for example, is $O(n^3)$ where n is the length of the input RNA sequence [117]. This notation is called **Big O** notation and is used in mathematics to describe the asymptotic behaviour of a function. For *mfold* this means the number of steps needed to fold a sequence grows in the order of n^3.

5.6.1 PSO computational complexity

For a stochastic algorithm like PSO, the complexity is harder to determine. Attempts have been made to analyse the complexity of a GA and it is concluded that the complexity cannot yet be formally defined for a GA [77].

Analysing the time complexity of classical PSO might be a more tractable problem than analysing a GA. Moving each particle through multidimensional space is a process that consists of a number of deterministic steps. These steps are calculation of the particle's new velocity and updating the particle's position, which are constant time or $O(n_x)$ calculations. Calculating the swarm's new position as well as updating the velocity are $O(n_s n_x)$ operations where n_s represents the number of particles and n_x represents the dimension of the particles. In each iteration of the algorithm, each particle's fitness should be calculated. The number of steps required to do this depends on the problem and, specifically, the objective function used. In the best case, the operation is a constant time operation for each particle $O(1)$, which translates to $O(n_s)$ for the swarm. The remaining factor which has a big influence on the complexity is the number of iterations the algorithm is executed. This number could be fixed or this number could be determined by stopping conditions like a convergence condition. If T is the total number of iterations completed by the computation, then the time complexity for classical PSO is in the order of $O(T n_s n_x)$.

5.6.2 SetPSO computational complexity

SetPSO has a few complications when it comes to complexity analysis. Like the GA, the individuals in the swarm of a SetPSO can have different lengths, and this determines the number of steps necessary to update each individual. The method of updating

the individual also has an impact. An optimising addition operator such as described in section 5.3.3, for example, will have greater complexity than a standard addition operator. The objective function used to calculate the fitness of an individual, in this case the mean free energy of RNA molecules, is dependant on the number of stems, l, contained in the individual, and is thus in the order of $O(l)$ complexity.

A discussion of complexity of all the subalgorithms of the SetPSO for this work is provided in this section. The SetPSO can be broken down into the following subalgorithms: stem enumeration, particle initialisation, particle evaluation, hill climbing and swarm updates. The complexity of each subalgorithm is listed below. The following symbols are used: n represents the length of the RNA sequence, n_s is the number of particles in the swarm, s is the number of stems in a particle, and c represents the number of stems in the close set. The complexity of each subalgorithm is:

- **Stem enumeration** is of $O(n^2)$, because each nucleotide in the sequence is matched to every other nucleotide.
- **Particle initialisation** is of $O(ln_s)$, because the number of particles to be initialised is n_s, each containing l stems.
- **Particle evaluation** is of $O(ln_s)$, because it is assumed that evaluation of a single particle is in the order of $O(l)$.
- **Swarm update** is of $O(lcn_s)$, because swarm update entails removing some of the current stems from each particle and adding new stems to each particle.

As with classic PSO, if the total number of iterations is T, the time complexity for SetPSO, excluding the stem enumeration operation at the start, is of the order $O(Tlcn_s)$.

To give an indication of the size of the search space, the size increases in an exponential relation to sequence length. It is estimated that the number of possible structures for an input sequence of length n is more than 1.8^n [20]. Because the search space grows exponentially with sequence length, the conformation of longer sequences is significantly harder to predict and requires significantly more computation time.

5.7 Conclusion

This chapter introduced a new set-based PSO algorithm, useful for combinatorial and set-based optimisation problems. The redefinition of the particles and the redefinition of the solution space were given in section 5.2.1. The new addition and subtraction operators, which need to work in the new set space, were discussed in sections 5.2.2, 5.2.3 and 5.2.4 after the solution space definition had been given in section 5.2.1.

Following a pseudo code algorithm for the new SetPSO in section 5.3, the important velocity update and position update methods were shown in sections 5.3.2 and 5.3.3 respectively. The three new parameters introduced by the new SetPSO algorithm were discussed individually in section 5.4.

Lastly, the computational complexity of the SetPSO algorithm was discussed. Determining the exact computational needs of a stochastic algorithm is not easy. A study of the algorithm's subprocesses and their complexities provides an indication of the overall complexity of the algorithm. The estimated computational complexity for SetPSO is of the order $O(Tlcn_s)$ and that of *mfold* is $O(n^3)$. This indicates that for large values of n, *mfold* might be at a disadvantage in terms of computational complexity. SetPSO should have an advantage over *mfold* in terms of computational complexity when $(T, l, c) < n$.

Chapter 6

Related work

An overview of previous work done in RNA secondary structure prediction is given in this chapter. The emphasis is placed on work which uses the mean free energy (MFE) minimisation technique to predict RNA secondary structures. Mean free energy minimisation methods are more comparable to this work than other methods such as comparative analysis.

6.1 Introduction

Most of the proposed algorithms for prediction of RNA secondary structure use a genetic algorithm (GA) as the optimisation algorithm. Among the work that proposes to use GAs to solve the problem are Shapiro and Navetta [82], Van Batenburg *et al.* [96], Benedetti and Morosetti [3], Shapiro *et al.* [83], Chen *et al.* [12], Zhang *et al.* [112] and Del Carpio *et al.* [18].

Three algorithms are discussed in this chapter. All three uses free energy (ΔG) as the objective to minimise, but each uses a different optimisation mechanism. In section 6.2, the *mfold* algorithm is discussed which uses dynamic programming (DP) to generate optimal solutions. P-RnaPredict is a modern algorithm which uses an evolutionary algorithm (EA) as its optimisation technique. P-RnaPredict is discussed in section 6.3. The final algorithm discussed also uses a modified PSO algorithm as its optimisation algorithm. HelixPSO is discussed in section 6.4.

6.2 *mfold*

The first implementation of a deterministic algorithm for predicting RNA secondary structure using an RNA thermodynamic model to evaluate the structure's energy, was done by Pipas *et al.* in 1975 [75]. The *mfold* program by Zuker *et al.* [62] [113] [114] [115] [116] [117] represents the latest developments in RNA secondary structure prediction using energy minimisation. *mfold* is a dynamic programming algorithm (DPA) with a complex thermodynamic model for free energy evaluation of conformations. Because the natural fold does not always correspond to the fold with the lowest energy [50] [51] [63], it is necessary to predict a number of suboptimal foldings, some of which will be more accurate than the lowest energy structure, although it is not possible to tell which structure is the most accurate without a known reference structure.

The *mfold* DPA uses INN-HB parameters [110], and adds modelling for some common RNA substructures like single base stacking energies, hairpin loops, interior loops and other energies. *mfold* has a helper program called *efn2* which re-evaluates the structures with a more complete thermodynamic model. This helper program includes improvements to the multi-branch loop thermodynamic model. *mfold* was also ported to the Microsoft Windows platform with a point and click interface. This port is called RNAStructure [64].

6.3 P-RnaPredict

P-RnaPredict is a parallelised version of RnaPredict and is still actively being developed [105]. RnaPredict has evolved to a level where prediction quality is comparable to that of *mfold* [106]. P-RnaPredict is a genetic algorithm (GA), more broadly classed as an evolutionary algorithm (EA). Evolutionary computation refers to methods by which evolution is simulated on a computer. The algorithms reliant on these methods are known as evolutionary algorithms (EAs), and include evolutionary programming (Fogel et al., [29]), genetic algorithms (Fraser, [31] and Holland, [46]), and genetic programming (Koza, [59]).

Generally, an EA uses a genetic representation of a solution to a problem which it optimises through a number of evolutionary steps. After an initial population of can-

didate solutions (individuals) are created, an evaluation function (objective function) determines the fitness of each individual. More fit individuals are allowed to reproduce offspring, through genetic operators, which are allowed into the next generation of individuals. Thus, natural selection occurs in the genetic population.

Selection operators include the elitism operator, which retains the best individuals in the population for the next generation, various crossover operators which is responsible for creating offspring from selected individuals, and a mutation operator which introduces random genetic material into a solution to enhance population diversity.

An evolutionary process takes place for a number of generations until the algorithm converges on an optimum solution or some other stopping criteria are met. The most fit individual at that point is taken to be the solution.

P-RnaPredict constructs a set, H, of RNA helices (stems) from which an individual is created (refer to section 7.3 for an explanation of how such a set can be constructed). Each helix in H has an index and the individual in P-RnaPredict is simply an ordered list of the indexes of the helices in H which makes up a RNA secondary structure.

P-RnaPredict uses either the individual nearest-neighbour (INN) or individual nearest-neighbour hydrogen bond (INN-HB) thermodynamic model as its objective function. See section 3.3 for a discussion on INN and INN-HB.

6.4 HelixPSO

Recently, another PSO approach to predicting RNA secondary structure has come to light. Geis and Middendorf proposed the HelixPSO [36]. HelixPSO uses a modified PSO and the ViennaRNA package to determine structures with minimum free energy.

Similar to P-RnaPredict, HelixPSO encode a structure as a permutation of indexes from the set of all helices, H. Similar to SetPSO, a particle moves with respect to a target position. Each particle i has an associated set of candidate target positions T_i and for each $t \in T_i$, a weight $w(t) > 0$. The relative weight of a position in T_i determines the probability that T_i is chosen as a target. T_i is initialized with a random position, i.e., a permutation that is generated randomly, and T_i is assigned a weight of 1.0. After each iteration of HelixPSO, each weight is decreased by multiplication with a parameter

$\rho, 0 < \rho < 1$. A position that has a weight less than a threshold, τ, is removed from T_i. Then the personal best position and either the global best or the neighbourhood best position are added to T_i with probability $c_1 \cdot r_1$ and $c_2 \cdot r_2$, respectively, where r_1 and r_2 are random numbers sampled uniformly within the range $[0, 1]$.

When a particle i has chosen its target position from the set T_i, the particle moves towards the target. This is done by performing a number of transpositions in the particle vector in such a way that the particle becomes more like the target.

The objective function used by HelixPSO is, however, not just a free energy calculation of a structure. The fitness function is a combination of each structure's free energy and the structure's similarity to a centroid structure. First the free energy component is computed. The energy component is a ratio between the structure's energy and the mfE, which is the structure with the lowest energy as computed by the Vienna RNAfold DPA. The second part is the agreement between the structure and the centroid structure which is computed by the Vienna package's centroid function. The complete fitness function is thus

$$Fitness(S) = \lambda * \frac{|S \cap C|}{|C|} + (1 - \lambda) * Min\left\{0, \frac{E(S)}{mfE}\right\}, 0 <= \lambda <= 1 \qquad (6.1)$$

where S is the potential solution, C is the centroid structure determined, and λ is a scaling weight which determines the influence of each component. By varying λ in equation (6.1), the influence of the components are determined. The resulting fitness is also in the range [0,1].

The centroid C is the structure with the smallest base pair distance to the structures in the thermodynamic ensemble of a sequence, of which the mfE is the member with the highest probability. The mfE is the lowest possible energy structure possible for the sequence and was determined exhaustively through the DPA. The actual solution determined by HelixPSO is however not the structure with the lowest energy; the lowest energy structure has already been found in the algorithm initialisation by the DPA! HelixPSO keeps a count of all the different structures that appears throughout the experiment and picks the structure that appears most frequent as the final solution. The reasoning behind this is that the structure that is formed most frequently is located in an easily accessible position in the energy landscape, and thus is a good candidate to be the native conformation.

6.5 Conclusion

This chapter described previous work done in the realm of RNA secondary structure prediction. The genetic algorithm approaches were mentioned first. *mfold*, the long time benchmark in RNA structure prediction was introduced in section 6.2. The current state of the art free-energy minimisation algorithms, namely P-RnaPredict and HelixPSO, where described in sections 6.3 and 6.4.

Chapter 7

RNA modelling

This chapter describes how the SetPSO algorithm can be used to predict the conformations of RNA molecules. A number of aspects of this SetPSO implementation are described next. The objective function used to determine the fitness of a particle is defined in section 7.1. Section 7.2 describes how a SetPSO particle can represent an RNA secondary structure. Section 7.3 provides the steps, called stem enumeration, used to generate the elements of the sets in SetPSO. The modified addition operator used in this implementation is discussed in section 7.4.

7.1 Objective function

The Gibbs free energy, or ΔG, of the RNA secondary structure is used as the objective function for the SetPSO experiments. Free energy is a thermodynamic potential which measures the process-initiating work obtainable from an isothermal, isobaric thermodynamic system, first described by Josiah Gibbs in 1870.

A strong correlation between the free energy of a conformation and the accuracy of the conformation exists [5] [6] [7] [62] [93]. As the free energy reduces, the stability of the structure increases.

The Vienna RNA package's energy calculation routines are used to compute ΔG [45]. The Vienna RNA package uses stacking energy rules and parameters from Mathews *et al.* [62], the same energy rules used by Zuker's mfold algorithm [115]. The stacking energy

model is discussed in section 3.3. The Vienna energy calculation routines are unable to deal with pseudo-knot motifs. Pseudo-knots are not considered in the experiments for SetPSO structure prediction. If in the future, the Vienna RNA package adds support for calculating pseudo-knot energy contributions, it will be possible to extend the SetPSO implementation to enable prediction of pseudo-knots.

7.2 Particle representation

A conformation can be seen as consisting of a collection of stems. A stem in turn consists of a set of stacked nucleotide pairs. This set of stacked pairs is formed by two or more base pairs $(i, j), ..., (i+n, j-n), 1 \leq n \leq m$, such that the ends of the pairs are adjacent, forming a ladder-type structure. Figure 7.1 shows an example of the stems that make up the known conformation of *Saccharomyces cerevisiae*'s 5S rRNA. Each individual therefore consists of a collection of stems and the remaining nucleotides in the string are considered unbound. Benedetti and Morosetti used this representation first in their GA algorithm for finding suboptimal RNA secondary structures [2]. Wiese and Glenn also used this representation in their GA implementation, RnaPredict [106].

Certain restrictions apply to forming stems. First, the stem should consist of at least three consecutive canonical base pairs that form a stack. Secondly, any loop that is closed by the stem should be at least three nucleotides (nt) long (refer to figure 7.2). The minimum loop lengths are chemical requirements for the stable formation of stems. Although stems with a shorter length than three nucleotides can occur, this does not often happen. The minimum stem length of three allows construction of most of the possible conformations, but reduces the search complexity enormously.

All the possible stems with a minimum length of three are enumerated (refer to section 7.3 for a description of stem enumeration) and put into the universal set U. Particles are initialised from the universal set.

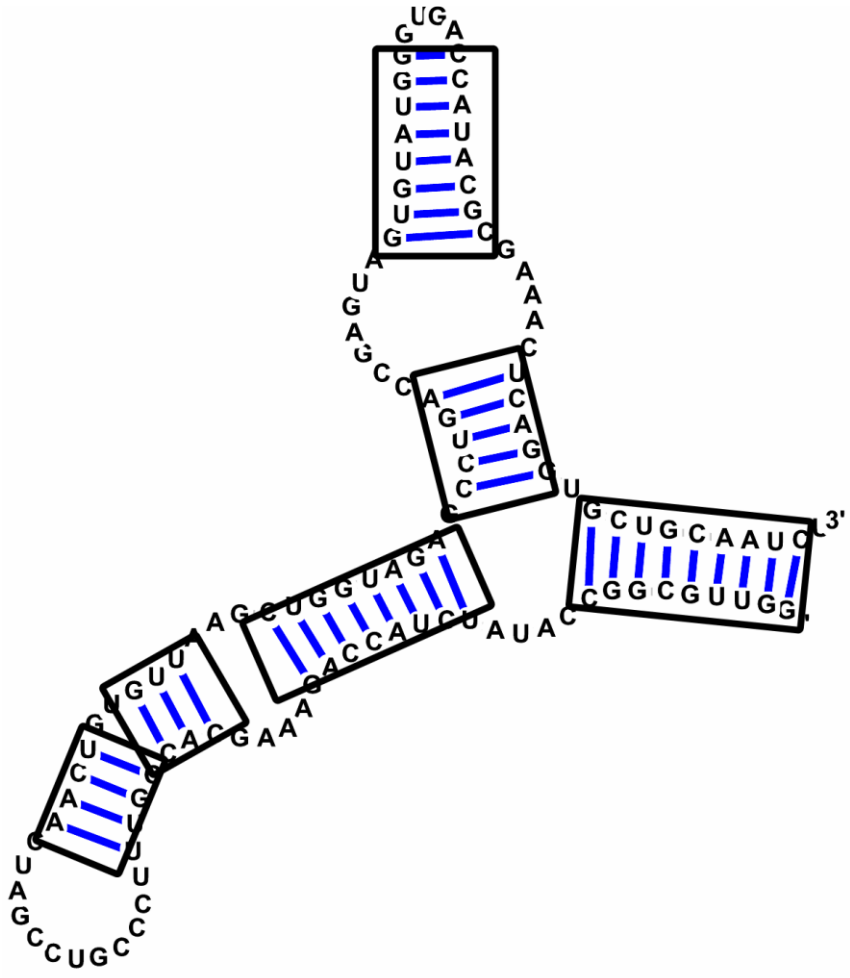


Figure 7.1: The boxes indicate the stems of a conformation. The conformation depicted is the known conformation of *S.cerevisiae*'s 5S rRNA.

7.3 RNA stem enumeration

The first step in structure prediction is to build a set U of all the potential helices (stems) which could form in a given RNA sequence under the thermodynamic model. First, all canonical base pairs for a given RNA sequence are iterated through, and the algorithm attempts to build a helix by stacking additional base pairs on top of existing ones. Consider an RNA sequence r of length n, and a canonical base pair (i, j), where $0 \leq i < j < n$. A base pair that does not meet the condition of $|j - i| > 9$ is discarded,

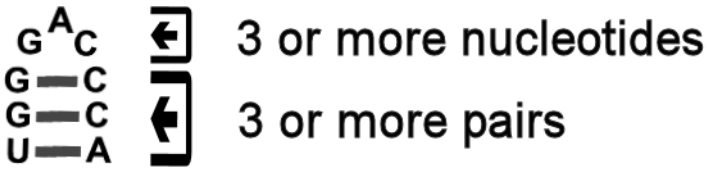


Figure 7.2: Restrictions on stem formation. The minimum stem length is 3 and the minimum loop length is 3.

as it could not be the base of a valid helix (see figure 7.2 which contains 9 nucleotides). If $(i+1, j-1)$ is a canonical base pair, this base pair is added to the potential helix h, and the next potential base pair $(i+2, j-2)$ is checked. This proceeds until the first non-canonical base pair is encountered, or until $(i+k, j-k)$ is reached, where $(j-k)-(i+k) \leq 3$. At this stage, if $k > 1$ the helix h is considered valid and is added to U.

With the complete helix set U, structure prediction then becomes a combinatorial optimization problem of picking a subset s from U. As helix generation stops at the first mismatched base pair, higher-order secondary structure elements are implicitly defined by the various bulges and loops outside of the helices stacked pairs. As a result, it is only necessary to determine the helices to account for all other secondary structure elements.

7.4 Optimised addition operator

The method that is used in the implementation of an optimised addition operator to add stems to the new position set is relatively simple. The operator introduces a bias regarding the stems it adds to the new position. The stems that are added first to the *close set* are those of the *lbest* and *pbest* positions. These are subsequently added to the new position first as well, before any new random stems are added to the current position set. This favours stems that are found in structures with better fitness.

The addition operator also serves to construct only valid RNA conformations. A lookup table is created after the stem enumeration is done. This lookup table indicates which stems are incompatible because of shared base pairs. When a stem is about to be

added to a set, a lookup is done in the lookup table to see if the stem will be in conflict with any of the existing stems in the set. When it is determined that no conflict will occur, the stem is added to the set.

7.5 Conclusion

This chapter explained how SetPSO is implemented in order to do predictions on RNA secondary structures. The objective function and particle representation were defined. The process of enumerating all the valid stems for the universal set, U, was outlined. Finally, the addition operator was described. The addition operator is responsible for ensuring that potential solutions are valid RNA structures.

Chapter 8

Experimental Approach

This chapter explains the experimental procedure followed in this work. The SetPSO RNA structure prediction algorithm is tested on various RNA sequences using different combinations of parameter values.

8.1 Introduction

This chapter details how the SetPSO experiments are done, what is measured, and to what the results are compared. Section 8.2 introduces the RNA sequences used in this study. Next, the measurements that are applied to quantify performance are explained in section 8.3. The results obtained from SetPSO are analysed and compared to the results of other RNA prediction algorithms. The method is described in section 8.4. Sections 8.5 and 8.5.1 and 8.6 describes the investigation that is done to determine the influence of the control parameter on various aspects of the SetPSO swarm. Finally, section 8.8 gives an overview of a useful visualisation tool called Parallel coordinates visualisation.

8.2 Sequences tested

In order to asses the feasibility of the SetPSO algorithm when used to predict the secondary structure of RNA, a number of RNA sequences are tested. The sequences and their known conformations are taken from the Comparative RNA Website [10]. Se-

quences were chosen from different species. The sequences were also chosen with different lengths to test the implementation's ability to predict conformations of different sizes. A short summary of each RNA sequence's attributes are given in tables. Each table contains the file name of the RNA sequence, its accession number, sequence length in nucleotides (nt), number of base pairs in the known structure (#BP) and the type of base pairs in the known sequence. RNA sequences of the following organisms are investigated: *Homo sapiens* 16S rRNA (954nt) in table 8.1, *Xenopus laevis* mitochondrial 12S rRNA (945 nt) in table 8.2, *Drosophila virilis* 16S rRNA (784 nt) in table 8.3, *Caenorhabditis elegans* 16S rRNA (697 nt) in table 8.4, *Aureoumbra lagunensis* 18S rRNA (468 nt) in table 8.5, *Haloarcula marismortui* 5S rRNA (122 nt) in table 8.6, *Arthrobacter globiformis* 5S rRNA (123 nt) in table 8.7, *Saccharomyces cerevisiae* 5S rRNA (118 nt) in table 8.8).

Table 8.1: *Homo sapiens* 16S rRNA details

Filename		d.16.m.H.sapiens.5.bpseq	
Accession number		J01415	
Length		954 nt	
# BP in known structure		267	
# BP by type			
GC	116	CU	2
AU	107	GG	1
GU	14	GG	1
CC	4	AA	5
CA	13	UU	4

Table 8.2: *Xenopus laevis* 16S rRNA details

Filename	d.16.m.X.laevis.bpseq
Accession number	M27605
Length	945 nt
# BP in known structure	257
# BP by type	
GC 126	CU 5
AU 95	GG 1
GU 12	GA 4
CC 5	AA 1
CA 6	UU 2

Table 8.3: *Drosophila virilis* 16S rRNA details

Filename	d.16.m.D.virilis.bpseq
Accession number	X05914
Length	784 nt
# BP in known structure	233
# BP by type	
GC 38	CU 1
AU 157	GG 0
GU 27	GA 4
CC 0	AA 1
CA 1	UU 4

Table 8.4: *Caenorhabditis elegans* 16S rRNA details

Filename	d.16.m.C.elegans.bpseq		
Accession number	X54252		
Length	697 nt		
# BP in known structure	189		
# BP by type			
GC	41	CU	4
AU	103	GG	2
GU	22	GA	5
CC	0	AA	4
CA	0	UU	8

Table 8.5: *Aureoumbra lagunensis* 18S rRNA details

Filename	b.I1.e.A.lagunensis.C1.SSU.516.bpseq		
Accession number	U40258		
Length	468 nt		
# BP in known structure	113		
# BP by type			
GC	59	CU	0
AU	33	GG	0
GU	15	GA	1
CC	0	AA	0
CA	2	UU	1

Table 8.6: *Haloarcula marismortui* 5S rRNA details

Filename	d.5.a.H.marismortui.bpseq
Accession number	AF034620
Length	122 nt
# BP in known structure	38
# BP by type	
GC 27	CU 1
AU 5	GG 0
GU 2	GA 1
CC 0	AA 0
CA 0	UU 2

Table 8.7: *Arthrobacter globiformis* 5S rRNA details

Filename	d.5.b.A.globiformis.1.bpseq
Accession number	M16173
Length	123 nt
# BP in known structure	39
# BP by type	
GC 12	CU 5
AU 1	GG 7
GU 2	GA 5
CC 3	AA 1
CA 2	UU 1

Table 8.8: *Saccharomyces cerevisiae* 5S rRNA details

Filename	d.5.e.S.cerevisiae.bpseq		
Accession number	X67579		
Length	118 nt		
# BP in known structure	39		
# BP by type			
GC	18	CU	2
AU	14	GG	0
GU	3	GA	0
CC	0	AA	0
CA	0	UU	0

8.3 Measurements

The metrics used to determine the accuracy of the predicted structures are the true positives (TP), sensitivity (SE) and specificity (SP). True positives is the number of correctly predicted base pairs in the candidate structure as compared to the known structure. The sensitivity is the ratio between TP and the number of base pairs in the natural fold, expressed as a percentage. Specificity is the ratio between TP and the number of base pairs in the candidate solution expressed as a percentage.

8.4 Accuracy of the structures and comparison to other algorithms

The SetPSO is compared to other free energy minimisation algorithms in terms of accuracy of predicted structures. These algorithms are the *mfold* DPA, RnaPredict EA and HelixPSO algorithms discussed in chapter 6. The results and accuracy comparisons are given in section 9.5.

All the results given in this work were generated using the *mfold* web server version 3.2. All the settings were left in the default state, including the percentage of suboptimality, which was set to 5%.

Results from HelixPSO are obtained by running experiments using the published HelixPSO code [37]. This algorithm has been modified and improved by the authors since the initial work published in [36].

Results for RnaPredict are obtained in the literature, mainly from [19] but also from other published articles [104] [105]. The results in [19] are more complete and include results on average accuracy over 30 samples. Results reported in [104] and [105] include the best results over several runs, not the average results over several runs. The authors of [104] and [105] sometimes report (and does comparisons to *mfold*) on the best result obtained by RnaPredict in terms of structure accuracy and not on the lowest energy structure. The flaw when using the best structure accuracy in reporting is that in practice the best structure cannot be known beforehand, except if the native structure is already known. This is different to the case where a lowest energy structure in a set can be determined since an evaluation function exist to do it. See [4] for a discussion on why reporting empirical *best* results for stochastic algorithms are problematic.

8.5 Investigating control parameters

The SetPSO algorithm introduced three new parameters, P_I or *entropy*, P_C or *closing probability* and P_R or *random add probability*. In order to investigate the influence of these three parameters, 64 experiments are conducted on each of five representative sequences. Each experiment consists of 30 samples. The experiments are run with the three parameters each taking on the values 0.2, 0.4, 0.6 and 0.8 in all possible combinations, giving rise to 64 experiments per sequence.

Other PSO parameters, e.g. the topology of the swarm and the number of particles, were not investigated. The Von Neumann swarm topology is used as it has been shown to be one of the best topologies to used in a wide range of problems [24] [56] [74].

Many studies have been done on the influence of swarm size [9] [26]. The number of particles, n_s, was set to 20. Empirical results have shown that a swarm size of between

10 and 30 particles is effective in finding optimal solutions.

The experiments were run under Suse Linux 10 with a SUN Java 5.0 VM. The source code for the experiments is part of the CIlib open source library [14].

SetPSO's ability to predict RNA secondary structures is investigated in chapter 9. The influence of the entropy parameter, P_I, is also investigated. It is also postulated that using a linear decreasing entropy weight, starting at P_I=0.9 and decreasing to P_I=0.1, throughout the course of an experiment, will lead to greater diversity of the swarm in the initial stages and more refinement of the solution in the later stages. The results of these experiments are shown in section 9.6.

8.5.1 Accuracy under linear decreasing entropy

Dynamically changing weights have been used in particle swarms to change the behaviour of the swarm during the experiment [67] [76] [95] [111]. In classic PSO, a linear decreasing inertia weight is an example of such a dynamically changing weight. The reason for changing the entropy weight dynamically is to try to influence the searching behaviour of the swarm. A large initial value for the entropy weight P_I means the particles should theoretically cover larger areas of the search space because larger entropy causes more disruption. As the experiment progresses, the value of P_I is decreased in order to try and exploit the search space around the best solutions.

The entropy weight is changed according to

$$P_I(t) = (P_I(0) - P_I(n_t)) \frac{(n_t - t)}{n_t} + P_I(n_t) \tag{8.1}$$

where $P_I(t)$ is the entropy at time step t, n_t is the maximum number of time steps for which the algorithm is executed, $P_I(0)$ is the initial entropy weight and $P_I(n_t)$ is the final entropy weight. Note that $P_I(0) < P_I(n_t)$.

Over the 700 iterations of each experiment, the entropy value was decreased from an initial value of $P_I = 0.9$ to $P_I = 0.1$.

The results of the linear decreasing entropy experiments can be found in section 9.6.2.

8.6 Investigating the influence of weights on swarm diversity

When particles are disrupted by removing elements from their sets, the particles are in a position to take on new elements which could potentially put the particles in a totally different part of the search space. Taken to the limit, this implies that the entropy, P_I, could remove all the elements from a particle set and the particle could get a new random position that does not share an element with any of its previous positions. The entropy can be seen as the inverse inertia of a particle. The fewer the elements that change in the particle, the more inertia it is said to have.

In an attempt to explain the searching behaviour of the SetPSO, the swarm diversity during experiments is investigated. The P_I parameter is tested with different fixed values as well as with a dynamically changing value.

8.7 Minimum Stem Length

For the previously described experiments, a constraint was placed on the minimum number of paired bases that constitute a stem. This minimum length constraint is set in order to reduce computational complexity at the expense of possible minor loss in accuracy of the predicted conformations.

At least two of the three shorter conformations have stems with two or less base pairs. These are *A. lagunensis* (which also contains a pseudo knot) and *H. marismortui*. Examination of the known conformations for these sequences as given in figures 9.4 and 9.3 confirms the presence of shorter stem lengths. Theoretically, more accurate conformations can be predicted for these structures when stems with length 2 are included. *S. cerevisiae* on the other hand can theoretically be predicted perfectly when minimum stem lengths of three base pairs are used, as seen in figure 9.6. Section 9.8 investigates the accuracy of SetPSO when reduced stem length elements are used.

8.8 Parallel coordinates visualisation

To better understand the influence of the parameters on the behaviour of the particle swarm, a visualisation technique known as parallel coordinates is used in the results chapter [48] [49]. Parallel coordinates is a common way of visualising and analysing high-dimensional data.

Essentially, the method entails drawing a vertical parallel line for each of the dimensions in the data. These lines are referred to as axes. Each axis (or dimension) is scaled independently so that all the axes are of equal length. A point in n-dimensional space is then represented as a polyline with vertices on each axis. The position of the vertex on the i-th axis corresponds to the i-th coordinate of the point. Another method of connecting the vertices is to draw curves through the vertices on the parallel axes. Dependencies between parameters can be identified more easily with the aid of these curves. Curves are used for visualisation in this chapter because the structured way in which parameter values where chosen causes the polylines to overlap. Curves space out better to allow all the lines to show.

8.9 Conclusion

The experimental approach was outlined in this chapter. The nine sequences that were tested were listed first. A method of using SetPSO to predict RNA secondary structures was discussed next. This included the particle representation, the objective function used, the swarm topology and the values of the parameters investigated. To summarise, the particle position is a set of possible stems. The objective function is the free energy of the structure as computed by the Vienna RNA package. The swarm topology used is the Von Neumann structure. The entropy weight, random add probability, and the close probability are tested with several values.

A parallel coordinates visualisation technique was described in section 8.8 in order to facilitate the investigation.

Chapter 9

Results and Comparisons

Several experiments were done in order to asses the performance of the SetPSO algorithm. This chapter shows and discusses the results of the experiments. The results are also compared to results found by other RNA prediction algorithms employing free energy minimisation.

9.1 Introduction

This chapter reports on the results obtained from running a number of experiments as described in chapter 8. Tables with summarised experimental results are given in this chapter at the relevant locations for the reader's convenience. Five representative sequences are discussed in this chapter. The complete tables of results for these sequences as well as the results of other sequences appear in appendix A. The five structures are *Xenopus laevis* - table 8.2, *Drosophila virilis* - table 8.3, *Aureoumbra lagunensis* - table 8.5, *Haloarcula marismortui* - table 8.6 and *Saccharomyces cerevisiae* - table 8.8.

The results for each sequence are then discussed and compared in section 9.4 to the known structure. The SetPSO results are also compared to HelixPSO, RnaPredict and the *mfold* DPA (dynamic programming algorithm) in section 9.5. *mfold* was discussed briefly in section 6.2. RnaPredict was discussed in section 6.3 and HelixPSO was discussed in section 6.4.

9.2 Initial results for SetPSO

The accuracy of the SetPSO algorithm in determining the secondary structures is shown first. These results were obtained by running multiple experiments with different values for the input parameters P_C, P_R and P_I. The tables with the complete experimental results are given in appendix A.

Table 9.1: Best fitness results obtained by SetPSO for the five sequences discussed

Sequence		P_R	P_C	P_I	ΔG (kcal/mol)	**Pairs predicted**	**Pairs correct** (%)
X. laevis	945nt	0.8	0.6	0.8	-192.3±5.0	242.6±6.0	67.0±10.7 (26.1%)
D. virilis	784nt	0.8	0.8	0.8	-113.0±4.1	255.1±5.1	30.7±6.7 (13.2%)
A. lagunensis	468nt	0.8	0.8	0.8	-128.3±1.8	128.6±3.0	47.1±7.6 (41.7%)
H .marismortui	122nt	0.8	0.6	0.8	-48.4±0	33±0	16±0 (42.1%)
S .cerevisiae	118nt	0.8	0.6	0.8	-53.4±0	40±0	28±0 (75.7%)

The results of the experiments of the representative sequences are reflected in table 9.1. The best fitnesses (ΔG) have been selected from each of the sequences' results table from appendix A. The corresponding P_C, P_R and P_I parameter values that gave the best result are shown.

An obvious first impression when looking at the summarised results in table 9.1 is that the longer sequences, *X. laevis* and *D. virilis*, show less accurate predictions than the shorter sequences. The longer sequences contain more unpredictable base pairs due to pseudo knots and non Watson-Crick base pairs. The shorter sequences show no deviation in the fitness of their predictions, suggesting that all the samples of the experiment converged to the same solution.

9.3 Accuracy

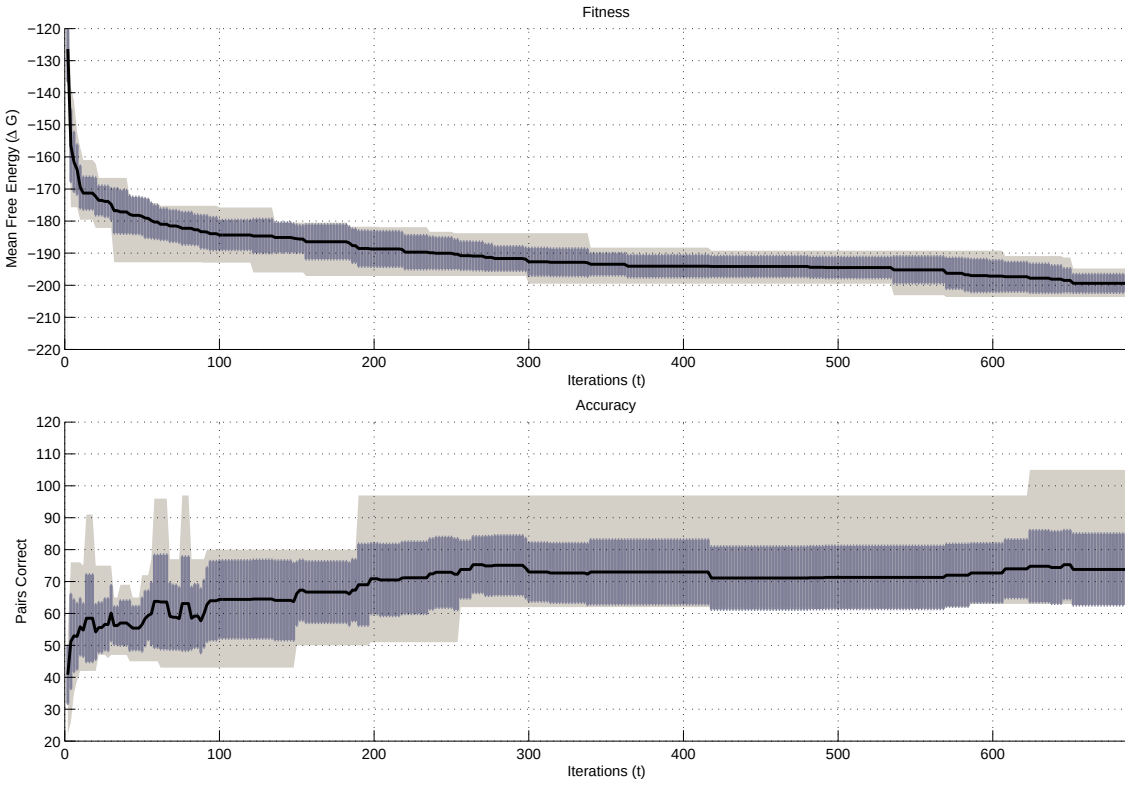
The first metric to be investigated is the accuracy of the secondary structure predictions.

A number of plots are included to show the average progression of the fitness[1] and

[1]One examiner suggested that the fitness of a population should always increase in an evolving

accuracy of the swarm (over 700 iterations) for each of the sequences. The plots indicate the fitness of the *gbest* particle as well as the corresponding accuracy of the *gbest* solution. The solid line represents the mean value of the fitness and accuracy respectively, while the dark grey area represents the standard deviation from the mean. The light grey area represents the minimum and maximum value over all the samples.

9.3.1 *X. laevis* accuracy



Graph 9.1: The fitness and accuracy of *X. laevis* over 700 iterations

population. While this is true, the objective function (ΔG) which represents the fitness of the population is minimized in this instance. This is acceptable and widely practised in the field of optimization.

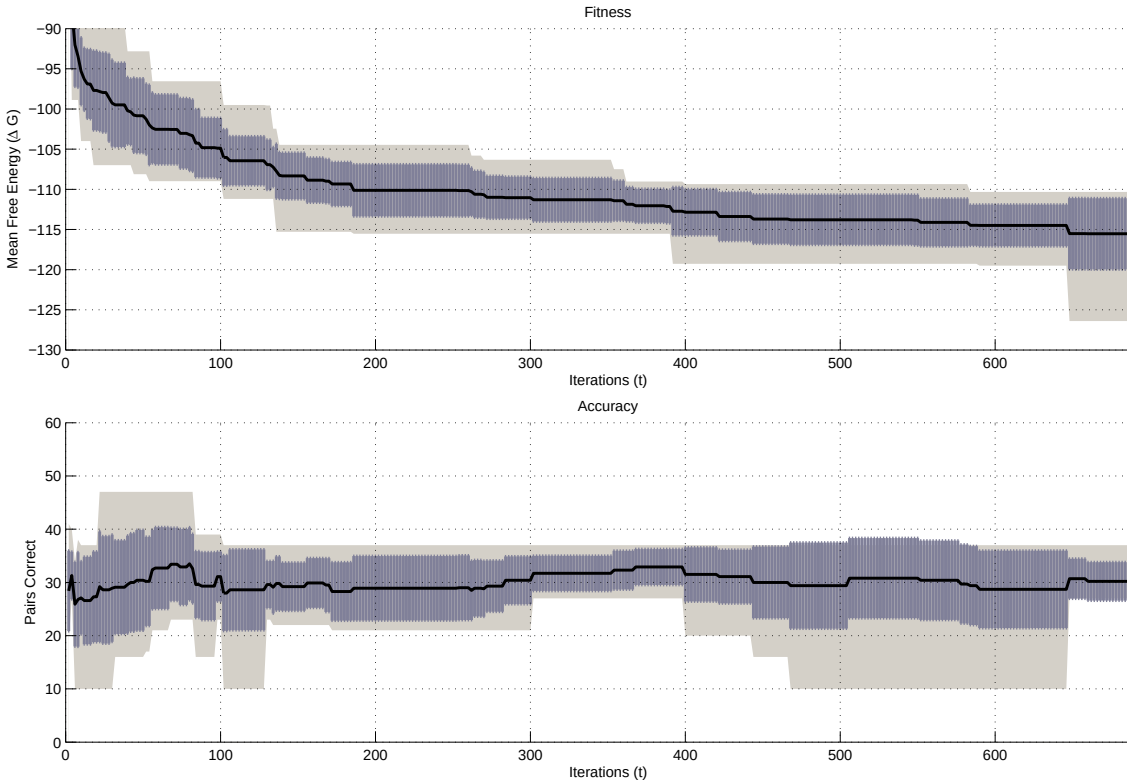
The two graphs in graph 9.1 show the improvement in mean free energy and the improvement in correctly predicted pairs over the 700 iterations of an experiment. This particular experiment was run with values of $P_R = 0.8$, $P_C = 0.6$ and $P_I = 0.8$. These parameter values resulted in the best swarm fitness of -192.3±5.0 as noted in table 9.1. The average number of base pairs predicted for the structures was 242.6±6.0 with a resulting accuracy of 67.0±10.7 correctly predicted pairs. However, the experiment with the best average accuracy after 700 iterations was recorded at 71.13±11.72 correct base pairs. This was with a parameter configuration of $P_R = 0.8$, $P_C = 0.6$ and $P_I = 0.6$. The average fitness for this configuration was -184.30±3.84 kcal/mol. See table A.1 in appendix A for the complete results set.

Generally, the experiments on the sequences show an increase in the accuracy of the predicted structure with an increase in the fitness of the solution. However, the accuracy does not continue to improve as the fitness improves for *X. laevis*. Remember that a lower free energy represents a better fitness. In this case, the accuracy sometimes decreases for a while (see graph 9.1 between iterations 300 to 600) while the mean free energy is decreasing. This observation can be attributed to the fact that the natural conformation of an RNA molecule does not always correspond to the absolute lowest energy state it can attain [50] [51] [63].

9.3.2 *D. virilis* accuracy

The fitness and accuracy graphs for the *D. virilis* sequence are illustrated in graph 9.2. These experimental runs were obtained with parameter values of $P_R = 0.8$, $P_C = 0.8$ and $P_I = 0.8$ which correspond to the best average fitness of the swarm. The fitness graph shows that the fitness was still decreasing when the 700 iteration limit was reached. The average fitness after 700 iterations was -113.0±4.1 with 255.1±5.1 base pairs predicted of which 30.7±6.69 were correct. The reason for the low accuracy will become apparent later in this chapter when the predicted structure is compared with the actual known structure.

The best accuracy obtained for *D. virilis* was with a parameter configuration of $P_R = 0.8$, $P_C = 0.6$ and $P_I = 0.4$. The average accuracy was 38.80±8.17 correct base pairs and the average fitness was -75.72±5.38. This is much lower than the best average



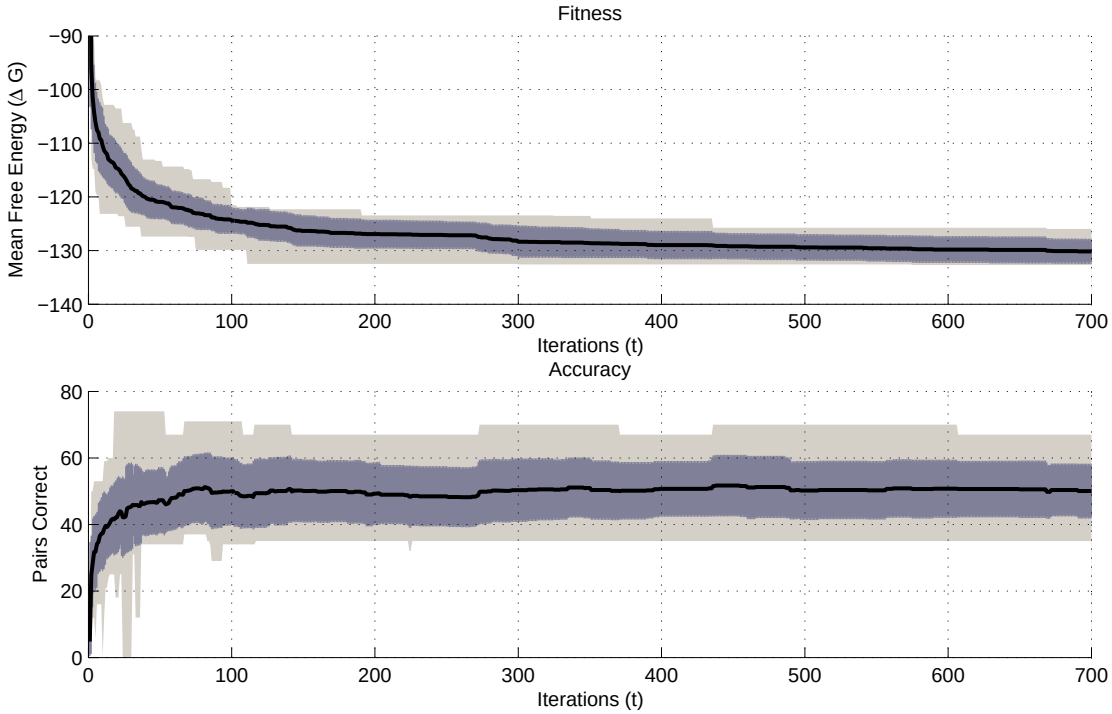
Graph 9.2: The fitness and accuracy of *D. virilis* over 700 iterations.

fitness found for this sequence.

The fitness and accuracy graphs show the same general trend as that of *X. laevis* in section 9.3.1. In fact, the most accurate structures for *D. virilis* in graph 9.2 were obtained even before the 100th iteration was reached, and the mean free energy was still relatively high.

9.3.3 *A. lagunensis*

The accuracy of the base pairs predicted in the experiments increased with shorter sequences. The *A. lagunensis* sequence is significantly shorter than the previous two sequences discussed and also showed a much better accuracy as can be seen from table 9.1. The average fitness of the swarm after 700 iterations was -128.3±1.8. The number



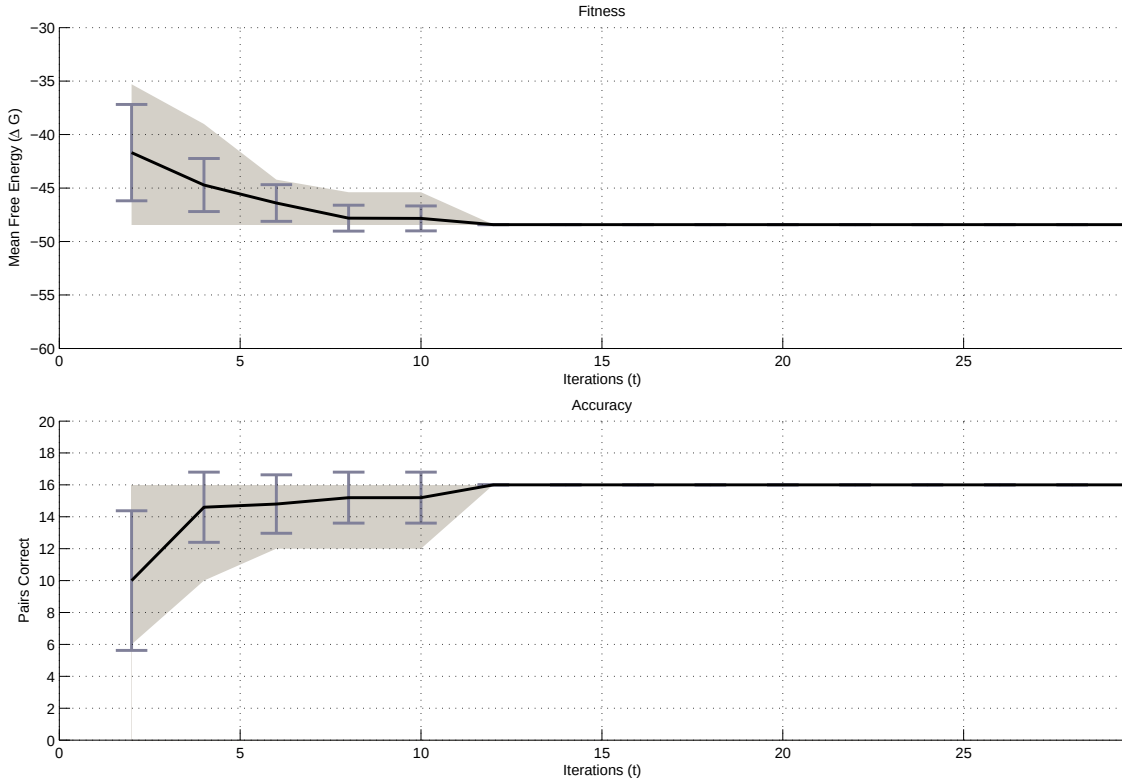
Graph 9.3: The fitness and accuracy of *A. lagunensis* over 700 iterations.

of base pairs predicted in the structure was 128.6±3.0 of which 47.1±7.6 were correct on average. This represents an accuracy of 41.7%. The parameter values that resulted in these best average fitnesses were $P_R = 0.8$, $P_C = 0.8$ and $P_I = 0.8$. The parameter values that resulted in the best average accuracy were, however, $P_R = 0.6$, $P_C = 0.2$ and $P_I = 0.8$. The average accuracy for these parameter values was 48.47±11.95. This is not

much better than the accuracy obtained from the best fitness configuration and it also has a larger deviation. See table A.7 for the complete results set.

The variation in fitness at the last iteration is much lower than that of the two longer preceding sequences. Some of the best structures were also predicted within the first 100 iterations when looking at the maximum shaded part of the accuracy graph.

9.3.4 *H. marismortui*



Graph 9.4: The fitness and accuracy of *H. marismortui* over 30 iterations.

The last two sequences are really short in comparison with the previously discussed sequences. The structure search space is much smaller with fewer possible stems that can form. The results showed that most of the parameter configurations gave the same results for these two short sequences. Table A.11 shows that 43 combinations out of the

possible 64 parameter combinations resulted in a fitness of -48.4±0. This means that every sample in the experiment found the same solution. The accuracy of the structure with this fitness is 16±0 bases correctly predicted. The total number of predicted bases is 33.

A notable exception, compared with the other sequences, is the number of iterations required to converge on the solution. In the case of *H. marismortui*, the *gbest* particle converged on a solution after a maximum of 12 iterations of the simulation, and never once changed afterwards. The search space for this sequence is so small that SetPSO has no problem in quickly finding the optimum solution.[2]

9.3.5 *S. cerevisiae*

The results for the *S. cerevisiae* sequence are very similar to those of *H. marismortui* in the previous section. The same fitness was reached in 53 of the 64 possible parameter configurations, namely ΔG = -53.4 kcal/mol. Further, *S. cerevisiae* only took a maximum of 8 iterations to converge to a solution. The results for all the parameter configurations are shown in table A.15.

The fitness and accuracy plots for the two short sequences in graphs 9.4 and 9.5 of the *gbest* particle show how quickly the solutions were found.

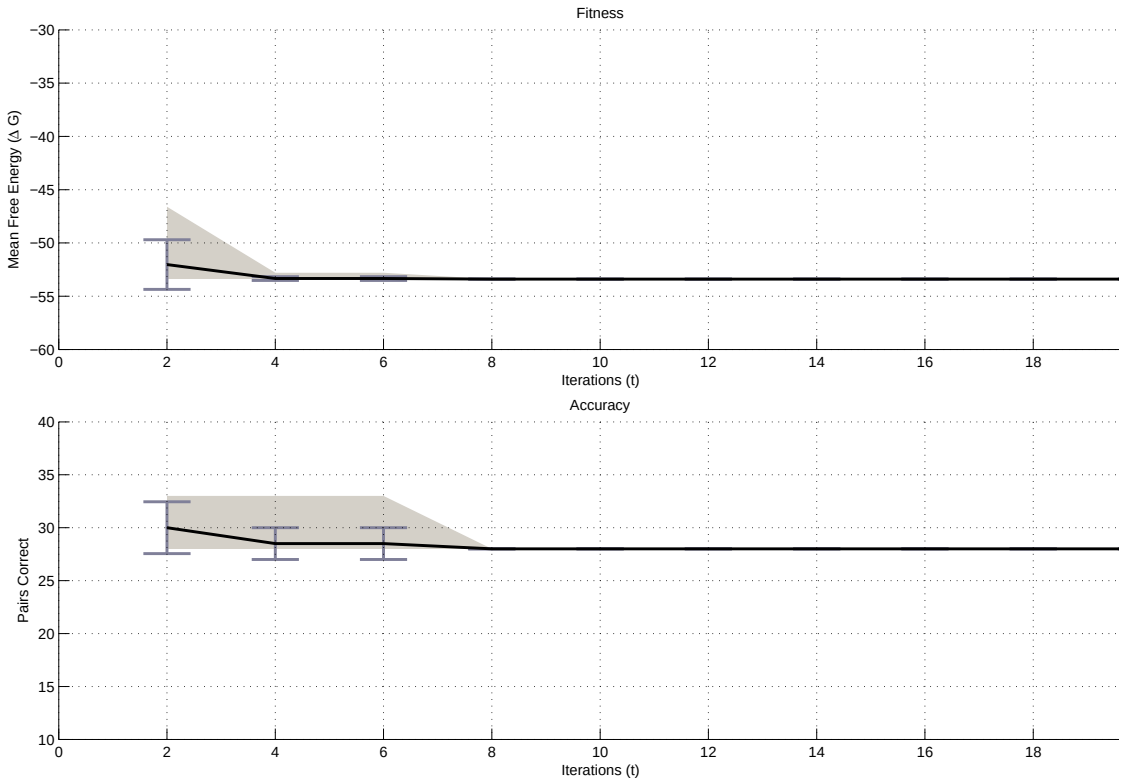
9.4 Comparison with known structures

Even though the base pairs are not entirely correctly predicted by the SetPSO, it is worth looking at the conformations and visualising them. Two conformations might only share 50% of the same base pairs, but their structures might be very closely related.

9.4.1 *Xenopus laevis*

Xenopus laevis is the longest sequence to be evaluated in detail. Table A.1 shows the results obtained from the experiments. The average free energy of the best fitness struc-

[2]The solution is optimum in terms of the objective function, i.e. the minimum fitness, and not in terms of the accuracy.



Graph 9.5: The fitness and accuracy of *S. cerevisiae* over 20 iterations.

tures was -192.3±5.0 kcal/mol. These structures contained 242.6±6.0 predicted base pairs of which 67.1±10.7 were correct, on average. That represents an accuracy of 26.1%.

Figure 9.1 shows a folding from one of the samples, compared to the known structure. The green lines represent the correctly predicted pairs while the blue lines show the unpredicted pairs or false negatives. The red lines are the false positive predictions. This particular predicted sample had 240 base pairs, of which 76 pairs were correct. 164 pairs are false positive predictions and there are 181 false negatives.

The known structure of *Xenopus laevis* contains 257 base pairs of which 24 are non-canonical base pairs. These 24 base pairs cannot be predicted with SetPSO. The known structure for *Xenopus laevis* also contains 4 pseudo knots which could not be predicted

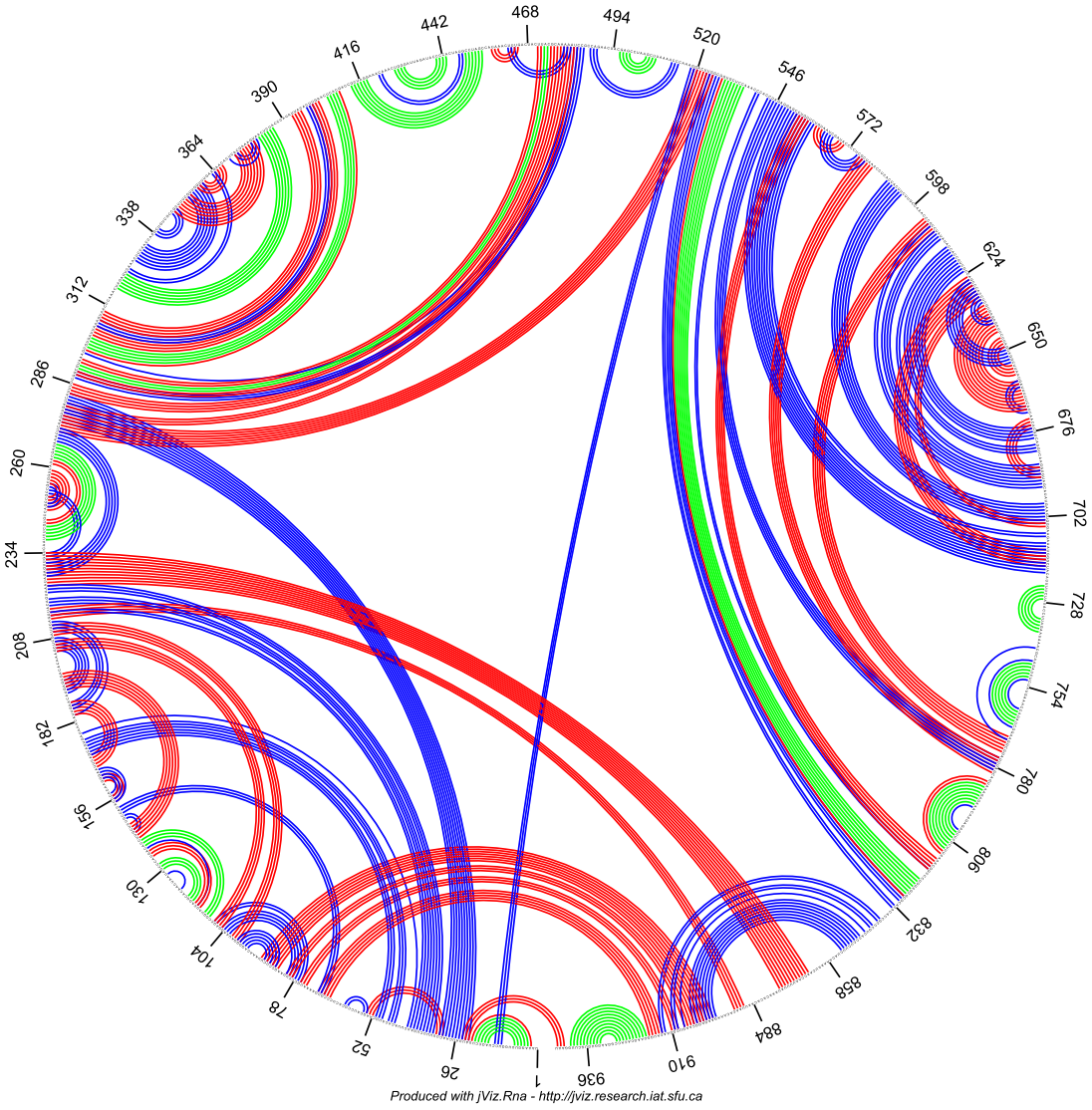


Figure 9.1: *Xenopus laevis* known and predicted structures, circular representation

because of the limitation of the free energy calculation function used.

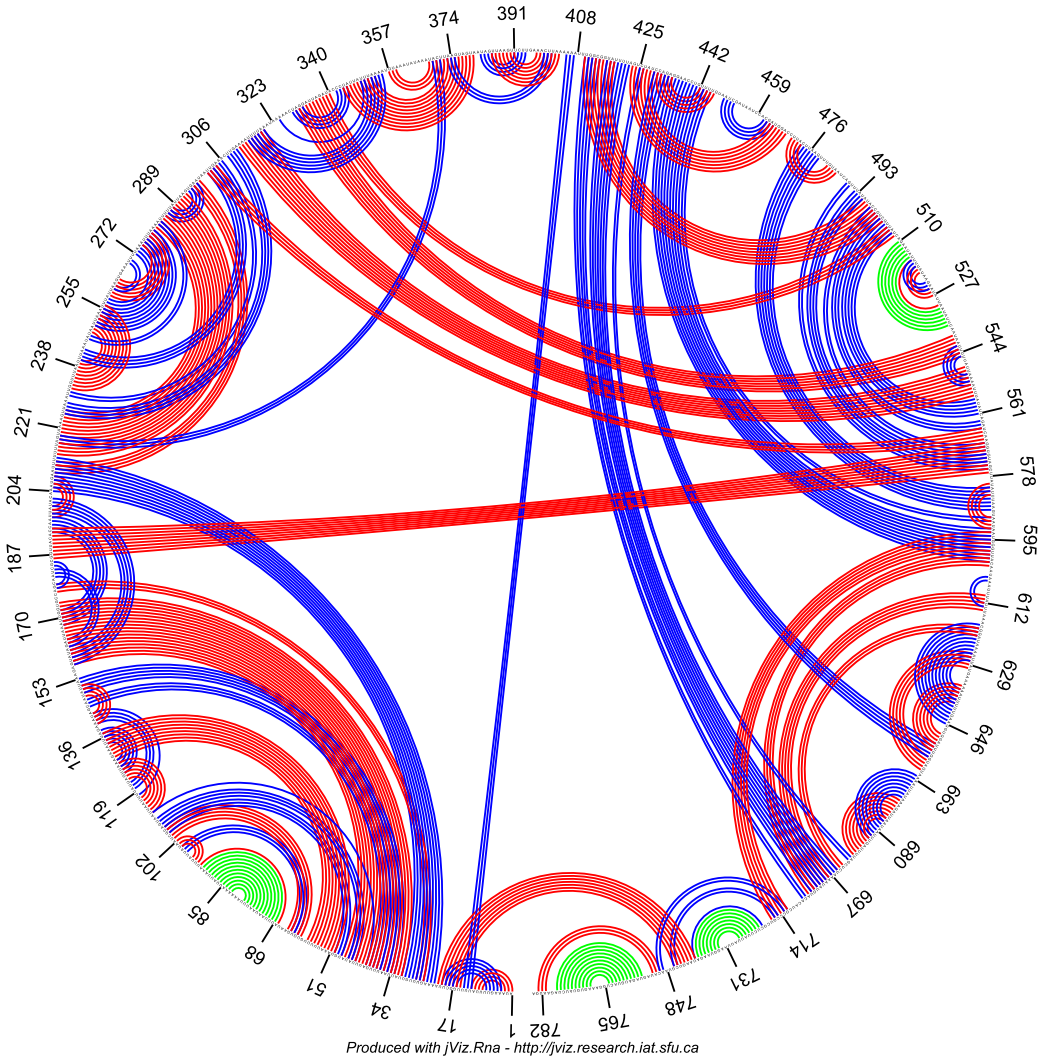


Figure 9.2: *Drosophila virilis* known and predicted structures, circular representation

9.4.2 *Drosophila virilis*

Drosophila virilis was the most difficult structure for SetPSO to predict. SetPSO managed to predict on average, across 30 samples, a structure with a minimum free energy of -113.0±4.1 kcal/mol. These structures contained 255.1±5.1 base pairs. Graph 9.2 shows how the average fitness of the population decreased over 700 iterations. The corresponding accuracy improved initially, but at iteration 90 it decreased suddenly and remained relatively poor, improving and deteriorating alternately. The accuracy of the average

structure predicted by SetPSO after 700 iterations was at a level where only 30.7±6.7 or 13.2% base pairs were correctly predicted out of 233 known base pairs.

Figure 9.2 shows a comparison between the known structure and one of the structures predicted by SetPSO. This particular structure had 256 base pairs, of which 34 were correct (the green lines), 222 pairs were false positives (the red lines) and 199 pairs were false negatives (the blue lines).

Eleven of the known base pairs are non-canonical base pairs and *Drosophila virilis* also contains 2 pseudo knots which could not be predicted at all.

9.4.3 *Aureoumbra lagunensis*

A graphical representation of a structure is a convenient means of getting an overview of its similarity or dissimilarity to other structures. A circular representation of the predicted and known structure for *A. lagunensis* is shown in figure 9.3 because it is a more compact representation for a longer sequence than a structural view.

Figure 9.3 shows the known conformation of *A. lagunensis* obtained from the Comparative RNA website, with the SetPSO predicted conformation as an overlay. The predicted conformation was taken from one of the experimental examples and contains 127 base pairs, of which 56 where correct. One noticeable feature of the known conformation is a pseudo-knot motif, indicated by some of the intersecting blue connecting lines.

Almost half the known base pairs are present in the SetPSO predicted conformation. Notice the similarity in the two structures between base number 90 and base number 286 and again between base number 358 and base number 404.

The SetPSO predicted structures had an average free energy of -130.2±2.0 with 127.3±2.9 base pairs. The accuracy of SetPSO predictions improves markedly with the shorter *A. lagunensis* sequence. On average 47.1±7.6 or 41.7% of the 113 known base pairs were correctly predicted after 700 iterations, as noted in table 9.1. The best run out of the 30 runs actually predicted 67 correct base pairs (59.3%). *A. lagunensis* contains only 4 non-canonical base pairs and a pseudo knot that could not be predicted because of limitations in the energy model used.

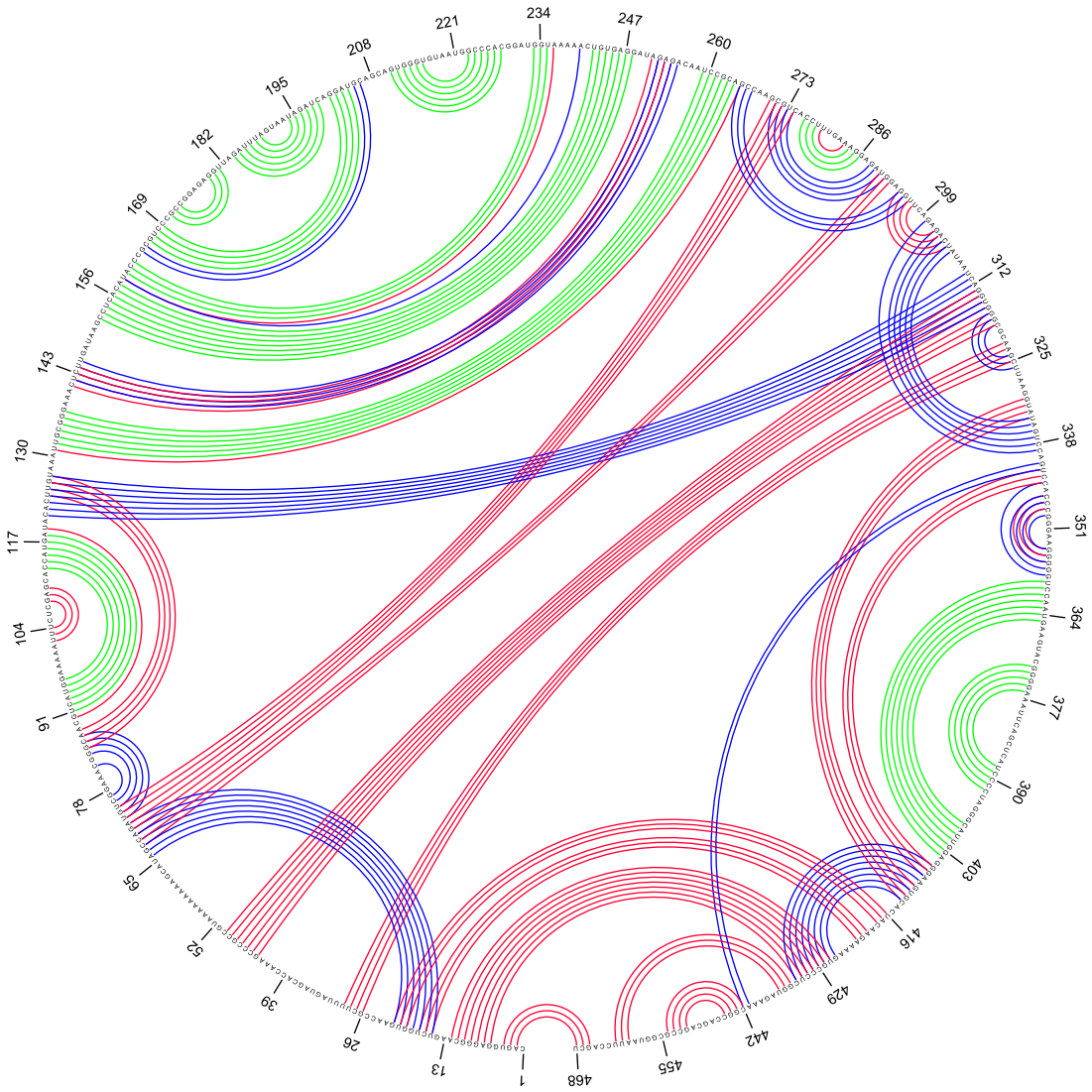


Figure 9.3: Known structure of *A. lagunensis*, circular representation

9.4.4 *Haloarcula marismortui*

For the *H. marismortui* sequence, SetPSO predicted a sequence, the same in all 30 runs, with a free energy of -48.4 kcal/mol. This structure had 33 base pairs. Remember that 16 of the 33 pairs of *H. marismortui* predicted by SetPSO were correct from table 9.1. That translates to a 42.1% accuracy for the 38 known base pairs. Although this is one of the shortest sequences, SetPSO did not do very well at predicting this structure.

Figure 9.4, the known conformation, and figure 9.5, the predicted conformation, provide visual confirmation of the differences in the conformations. The outermost stem (nearest the 3' and 5' ends) was predicted completely correct. The other features, e.g. the internal loop and the number of stems in the conformation were predicted less accurate. The structures differ to such an extent that the combined visualisation technique used for the other sequences does not give a clear view of the two foldings.

A possible reason for the lower accuracy of the predicted conformation could be the fact that a minimum stem length of three base pairs was set to limit the number of stems in the universal set, hence the complexity of the problem. The known conformation contains two stems with only two base pairs, as can be seen in figure 9.4. The influence of shorter stems is investigated in section 9.8. *H. marismortui* also contains three non-canonical base pairs that can not be scored by the objective function used.

9.4.5 *Saccharomyces cerevisiae*

Again, for this short sequence, the same structure was predicted in all 30 runs. The structure had a free energy of -53.4 kcal/mol and contained 40 base pairs. *S. cerevisiae* had the most correctly predicted base pairs of all the sequences tested, with 28 out of the 40 predicted pairs correctly predicted. Again, the outermost stem was predicted correctly.

Figure 9.6 shows the known and predicted structures of *S. cerevisiae*. The green bands indicate the pairings in both foldings, the blue bands indicate pairings only on the known structure (false negatives) and the red bands indicate pairings only in the predicted structure (false positives). Notice the similarity between the structures. In this case, the motifs such as the multi-branch loop, internal loops, and hairpin loops are all present, although not always in the correct form. The known *S. cerevisiae* contains only canonical base pairs, thus no stem was impossible to predict by means of the SetPSO.

9.5 Comparison with RnaPredict, HelixPSO and *mfold*

The accuracy of the predictions made by SetPSO is compared to the predictions made by *mfold*, RnaPredict and HelixPSO which also use free energy minimisation as an objective function.

Table 9.2: Average sensitivity of SetPSO, RnaPredict, HelixPSO and *mfold*

			RnaPredict			
Sequence		SetPSO	INN-HB	INN	HelixPSO	*mfold*
H. sapiens	954nt	20.6	18.0	17.2	25.5	35.7
X. laevis	945nt	26.1	25	25.1	30.4	36.7
D. virilis	784nt	13.2	12.7	16.5	18.7	15.9
A. lagunensis	468nt	41.7	36.7	41.7	38.4	53.1
H. marismortui	122nt	42.1	42.1	42.1	76.3	76.3
S. cerevisiae	118nt	75.7	89.2	75.7	78.8	89.2

Table 9.2 summarises the average solution accuracy (sensitivity) as predicted by SetPSO, RnaPredict, HelixPSO and *mfold*. SetPSO performed the same as RnaPredict except for *D. virilis* and *S. cerevisiae* where at least one of RnaPredict's thermodynamic models outperformed SetPSO.

HelixPSO performed almost as well as *mfold* on most sequences and also outperformed SetPSO in most cases. SetPSO and HelixPSO uses the same energy model as *mfold*, thus SetPSO and HelixPSO will not be able to find a structure with a lower energy than *mfold* does, simply because *mfold* performs an exhaustive search over the structure space and returns the absolute lowest energy structure. SetPSO can only approach *mfold* in terms of free energy, never surpass it. But this does not mean the algorithms can not predict a more accurate structure as the most accurate structure often does not have the lowest energy.

SetPSO scores lower on the sensitivity metric than *mfold* and HelixPSO on the *H. marismortui* sequence in particular. In section 9.8.2, this low accuracy of SetPSO is addressed, and a more favourable result is obtained.

From table 9.2 it seems that the HelixPSO has a better objective function than

SetPSO, in the combination of free energy minimisation and similarity to the centroid structure. All three of the stochastic algorithms performed worse than the DPA though. The structure search space is probably too convoluted for the stochastic optimisation algorithms to consistently find the lowest energy structures.

9.5.1 *Xenopus laevis*

The *mfold* results are presented in table A.3. The first column gives the free energy of the structure optimised by *mfold*. The second column gives the energy of the conformation after being evaluated by *efn2*. The third column indicates the number of base pairs predicted. The fourth column shows the number of correctly predicted base pairs while the last column displays the percentage of correctly predicted base pairs.

The structure with the lowest energy predicted by *mfold* had a mean free energy of -250.6 kcal/mol and contained 249 base pairs of which 92 pairs were correct, giving an accuracy of 36.7%. However, the most accurate structure is not the structure with the lowest energy. Scanning down the list, a structure with -248.0 kcal/mol and 245 base pairs had the most correct base pairs at 113 or 45.0%. It is only possible to tell that the structure is the most accurate because the natural conformation for *X. laevis* is known.

On average, SetPSO is not as accurate as *mfold*. SetPSO managed to predict a structure with 240±3.9 base pairs, of which 73.8±11.2 or (29.4%) were correct. Refer to table 9.1 for the SetPSO results. The most accurate structure of the 30 samples in the experiment contained 105 correct base pairs. This is almost as good as the best structure for *mfold*, which contained 114 correct base pairs.

9.5.2 *Drosophila virilis*

mfold predicted quite a number of structures within the 5% threshold limit for *D. virilis*. The lowest energy structure had a mean free energy of -146.3 kcal/mol with 236 base pairs of which 37 were correct (out of 233 known base pairs). This translates to an accuracy of 15.9%. However, the two most accurate structures only appear much lower in the energy rankings in table A.6, which summarises the results for the *mfold* DPA. The most accurate structures both contained 82 correct base pairs, which translates to

a 35.2% accuracy.

SetPSO managed to predict structures with an average accuracy of 31.1±3.8 correct base pairs (refer to table 9.1). The most accurate of the samples after 700 iterations contained 37 correct base pairs. The most accurate structure overall was found between iteration 22 and iteration 82, while the swarm still had a lower overall fitness. This structure contained 47 correct base pairs or a 20.2% accuracy. Refer to graph 9.2 and notice the light grey area between iterations 22 and 82 which illustrates the maximum accuracy of the experiment.

9.5.3 *Aureoumbra lagunensis*

The *mfold* DPA generally made very good predictions on the *A. lagunensis* sequence. The folding with the lowest free energy had a $\Delta G = -142.35$ for the *efn2* evaluation function. Refer to table A.10 for the results of the SetPSO predictions. This sequence consisted of 128 base pairs, of which 60 were correct, with an accuracy of 53.1%. The most accurate structure, however, was the 12th structure in the list, which consisted of 133 base pairs of which 74 were correct. That is a 65.5% accuracy rate.

SetPSO predicted structures with an average accuracy of 47.1±7.6 base pairs which translates to an accuracy of 41.7%. Of the 30 samples, however, the most accurate structure had 69 correct base pairs, giving an accuracy of 61.1%. This is not quite as good as the best *mfold* conformation. Figure 9.7 illustrates the circular structure representation of the *mfold* prediction with the lowest energy, compared to the known structure. Part of the structure was predicted very faithfully but the thermodynamic model of *mfold* failed to predict accurately in the longer range pairings.

9.5.4 *Haloarcula marismortui*

For a short sequence like *H. marismortui*, the 5% threshold limit of *mfold* returned only 1 conformation as shown in table A.14. This singular structure had a mean free energy of -59.5 kcal/mol and contained 34 base pairs, of which 29 were correctly predicted. That is an accuracy of 76.3%. Figure 9.8 illustrates the actual structure that *mfold* predicted. The results contained a number of stems with only 2 base pairs and 1 base pair contact

that is not part of a stem. The overall structure is, however, much closer in form to that of the known structure of *H. marismortui* (refer to figure 9.4).

Because the structure that SetPSO predicted, as illustrated in figure 9.5, had the wrong shape, it predicted far fewer correct base pairs than *mfold* managed to predict.

The *mfold* results for *S. cerevisiae* contained 2 structures. The structure with the lowest energy was also the most accurate. This structure had a mean free energy of -53.5 kcal/mol and contained 41 base pairs, of which 33 were correct. This gives an accuracy of 89.2%. Figure 9.9 illustrates the secondary structure of *S. cerevisiae* as predicted by *mfold* and compared to the known structure. This structure compares favourably with the known structure of *S. cerevisiae*. There are only 8 false positive predictions and 4 false negative predictions for *mfold*.

SetPSO predicted, in all cases, a structure with 40 base pairs and an accuracy of 75.7% which translates to 28 correct pairs. This is, incidentally, the same as the second structure predicted by *mfold* (refer to table A.18 and table 9.1).

9.5.5 *Saccharomyces cerevisiae*

The *mfold* results for *S. cerevisiae* contained 2 structures. The structure with the lowest energy was also the most accurate. This structure had a mean free energy of -53.5 kcal/mol and contained 41 base pairs, of which 33 were correct. This gives an accuracy of 89.2%. Figure 9.9 illustrates the secondary structure of *S. cerevisiae* as predicted by *mfold* and compared to the known structure. This structure compares favourably with the known structure of *S. cerevisiae*. There are only 8 false positive predictions and 4 false negative predictions for *mfold*.

SetPSO predicted, in all cases, a structure with 40 base pairs and an accuracy of 75.7% which translates to 28 correct pairs. This is, incidentally, the same as the second structure predicted by *mfold* (refer to table A.18 and table 9.1).

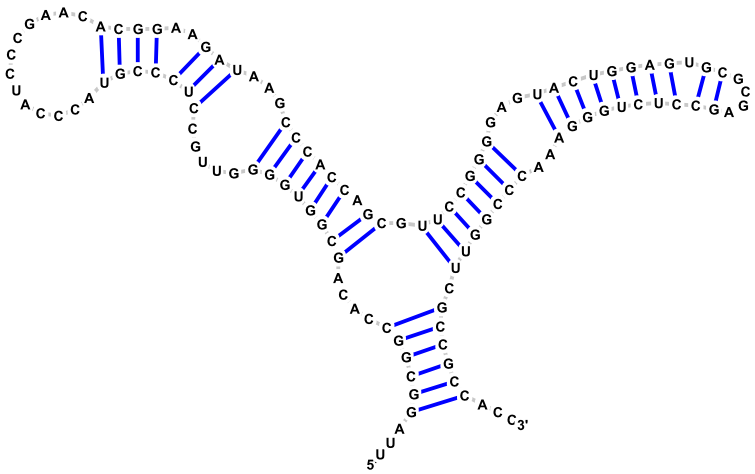


Figure 9.4: Known structure of *H. marismortui*, structural representation.

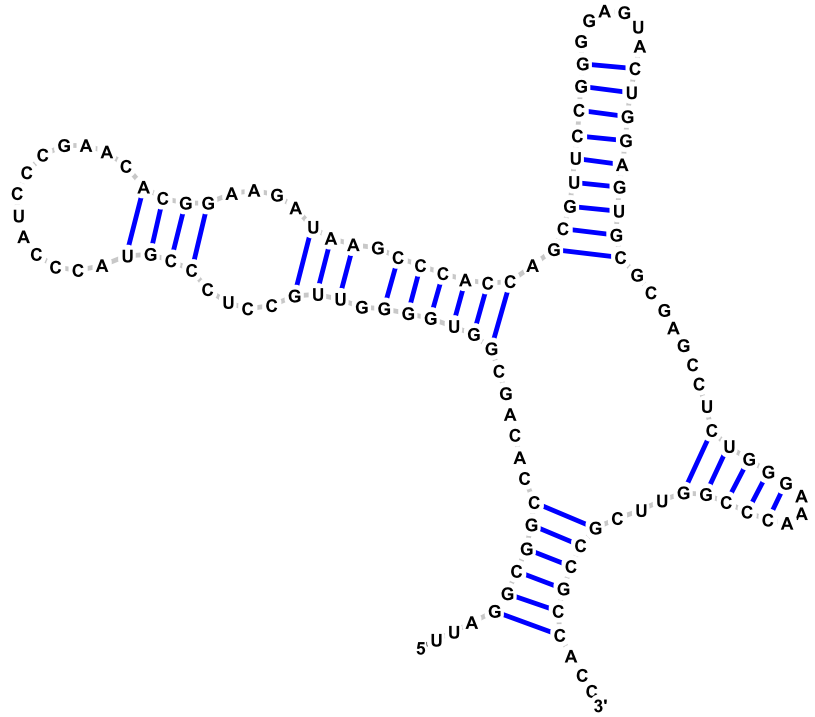


Figure 9.5: SetPSO predicted structure of *H. marismortui*, structural representation

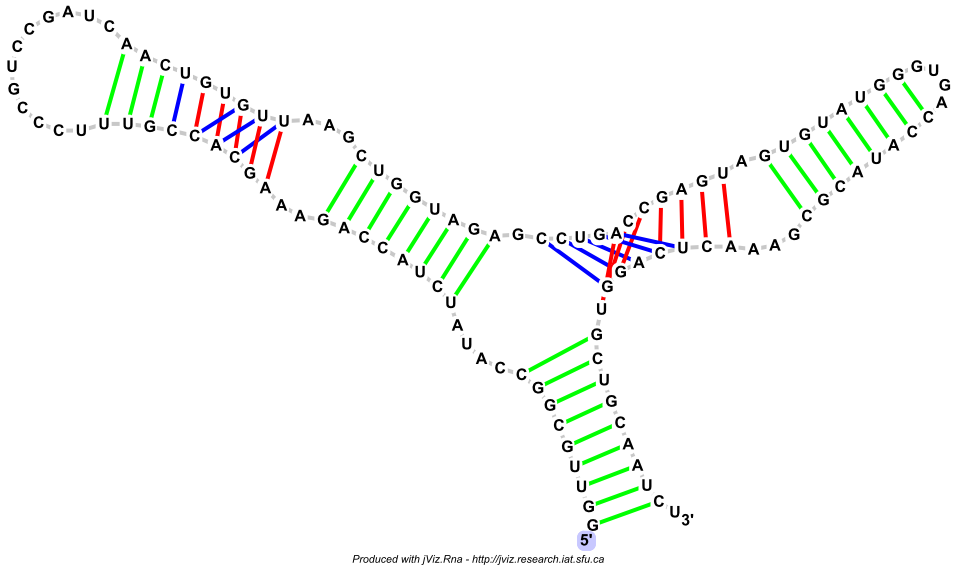


Figure 9.6: Combined known and SetPSO predicted structure of *S. cerevisiae*, structural representation

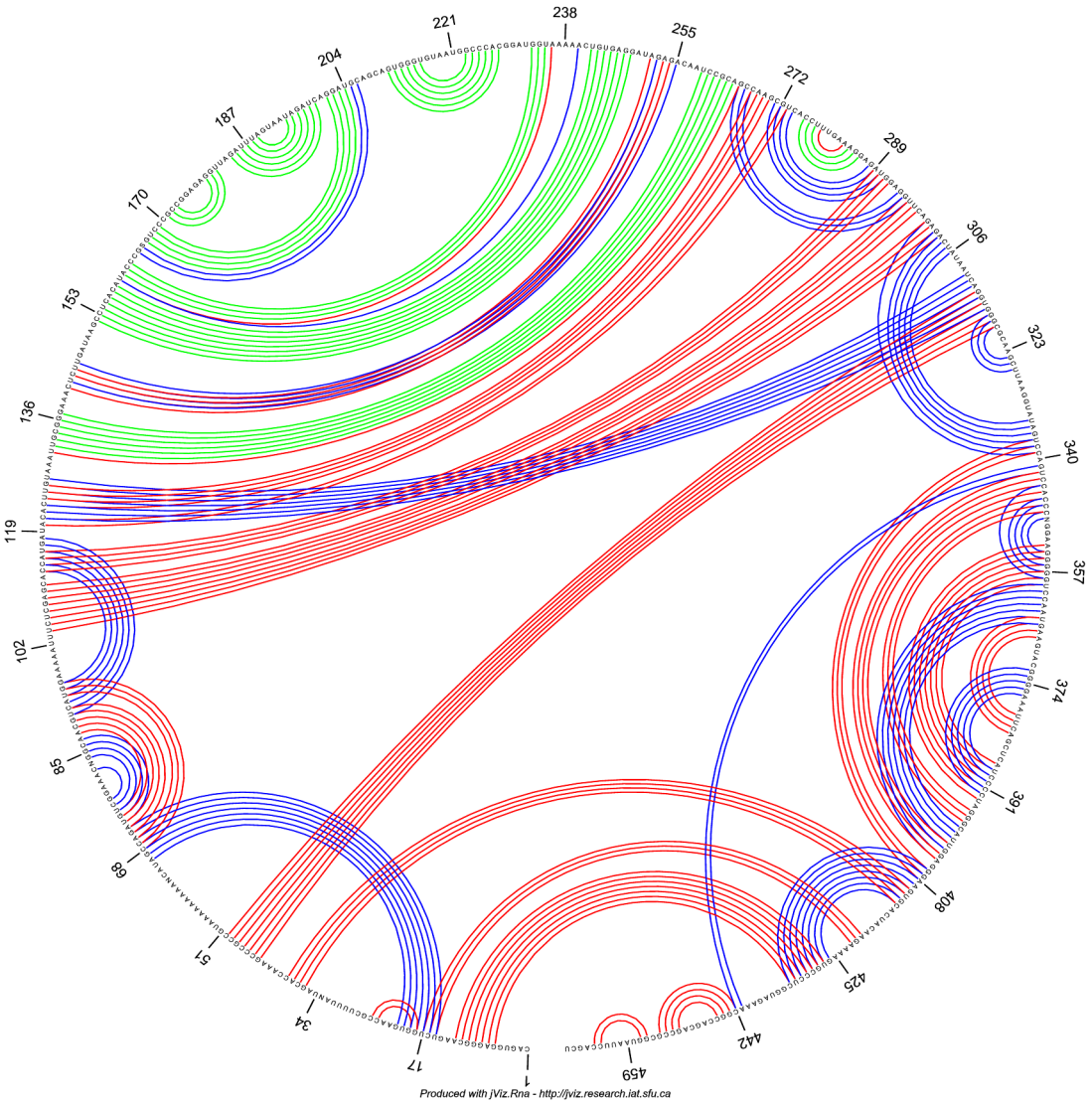


Figure 9.7: *A. lagunensis* structure predicted by mfold, using a circular representation

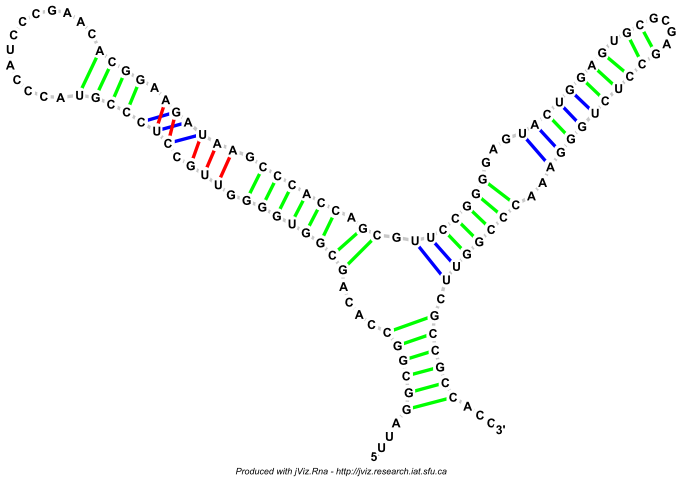


Figure 9.8: *H. marismortui* structure predicted by mfold, using a structural representation

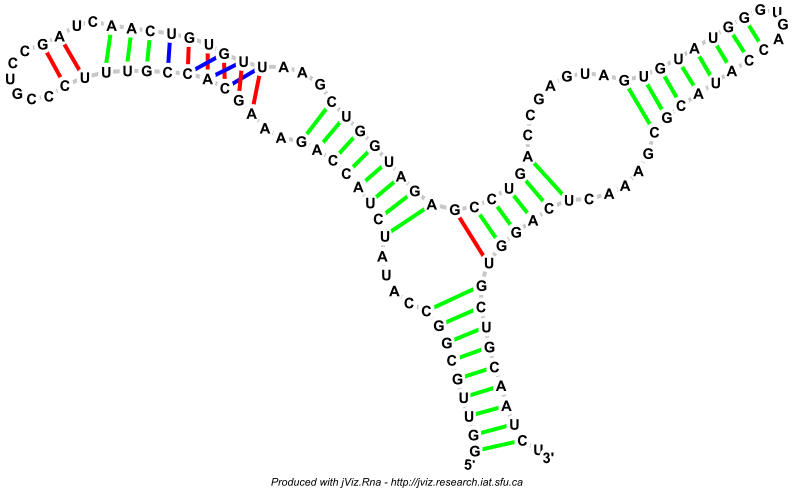


Figure 9.9: *S. cerevisiae* structure predicted by mfold, using a structural representation

9.6 Investigating the influence of SetPSO control parameters

The parameters used to tune a model or algorithm usually have a big impact on the performance of the model or algorithm. The SetPSO algorithm is no different. The three most important parameters of the SetPSO algorithm are the *closing probability* (P_C), *random add probability* (P_R), and the *entropy* weight (P_I), introduced in section 5.4. This section explores the influence that these parameters, especially the *entropy* weight, have on the performance of the SetPSO algorithm when predicting RNA secondary structures. Because RNA secondary structures with the best fitness in SetPSO does not correspond to the most accurate structures, the parameter investigation considers both the fitness and accuracy attained in the experiments.

Remember that the entropy weight determines the amount of disruption caused to particles (individuals) in the swarm. The higher the entropy, the more likely it is that elements will be removed from the set that constitutes a particle in the SetPSO. It is postulated that the *entropy* weight influences the diversity of the swarm. The larger the value of the entropy weight, the more elements are removed in each iteration, which leaves space for other elements to be added to the particle's set, thus increasing diversity in the swarm.

9.6.1 Accuracy under constant entropy

From section 9.2, table 9.1 and the results given in the tables in appendix A, it is clear that higher values of P_R and P_I resulted in better average swarm fitness. But what is the influence of P_C and are there any relationships between the control parameters? In order to gain some initial insight into the influence of the control parameters, a parallel coordinates plot of the 64 parameter combinations for the *X. laevis* sequence is shown in figure 9.10. The first three axes represent P_R, P_C and P_I in that order. The last axis represents the fitness values obtained from the experiment. The fitness is also known as the dependent variable. The axes have the lowest value at the top and the largest value at the bottom, thus the best fitness is at the top and the worst at the bottom.

The curves in figure 9.10 are of different colours. Each curve is coloured according to

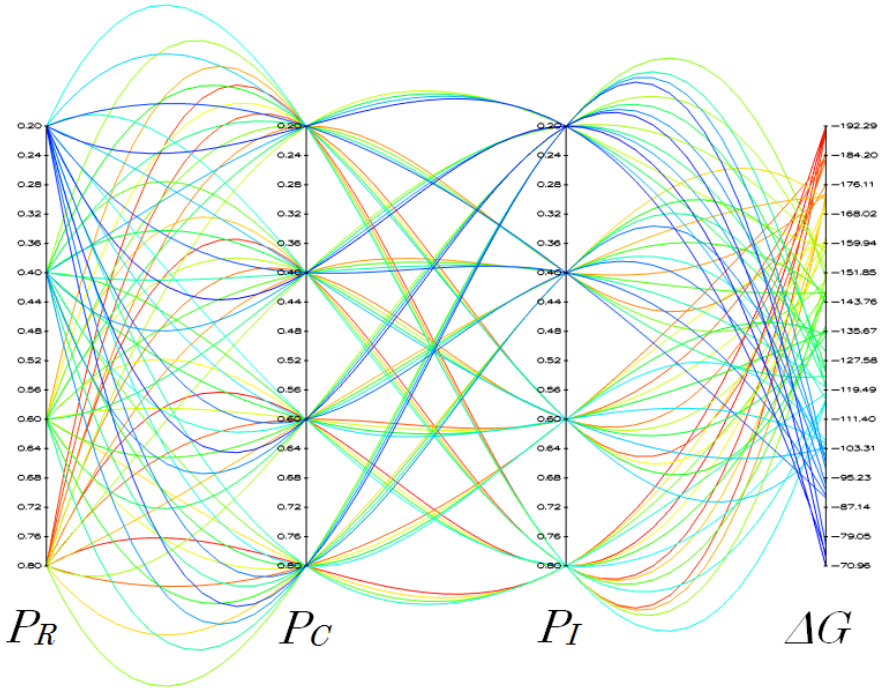


Figure 9.10: Parallel coordinate visualisation for parameter influence on *X. laevis* fitness. The axes represent P_R, P_C, P_I and ΔG.

the fitness value the curve represents. The red end of the spectrum represents the best fitness while the blue end of the spectrum represents parameter values that map to the worst fitness. Note the points on each axis where the red and dark blue curves cross; All the red curves cross the first axis at $P_R = 0.8$. This means that all the best solutions have $P_R = 0.8$ in common. All the dark blue curves cross at $P_R = 0.2$ which means all the worst fitness solutions have $P_R = 0.2$ in common.

In order to emphasise these best and worst patterns, the best fitness and accuracy patterns are highlighted in figure 9.11 and the worst fitness and accuracy patterns are highlighted in figure 9.12. The best fitness is at the top of the fitness axis (lowest value) while the best accuracy is at the bottom of the accuracy axis (higest value). The top 20% of the patterns are grey and the top 5% of the patterns are coloured. The top pattern is shown as a red dotted line.

Figure 9.11 shows that the three best fitness patterns have values of $P_R = 0.8$ and $P_I = 0.8$ in common, whereas the value for P_C varied. The three worst fitness patterns

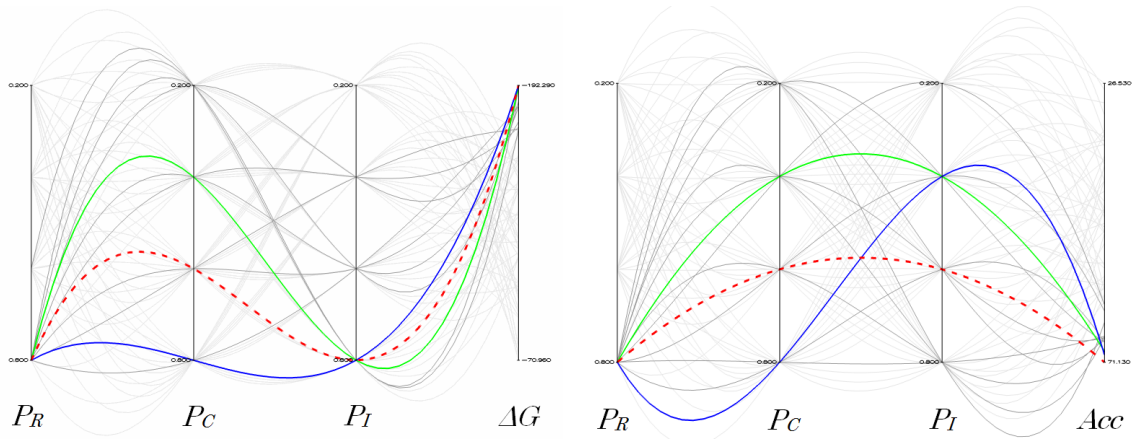


Figure 9.11: Curves mapping to top 20% fitness values on the left and the top 20% accuracy values on the right for *X. laevis*. The top 5% are shown in colour.

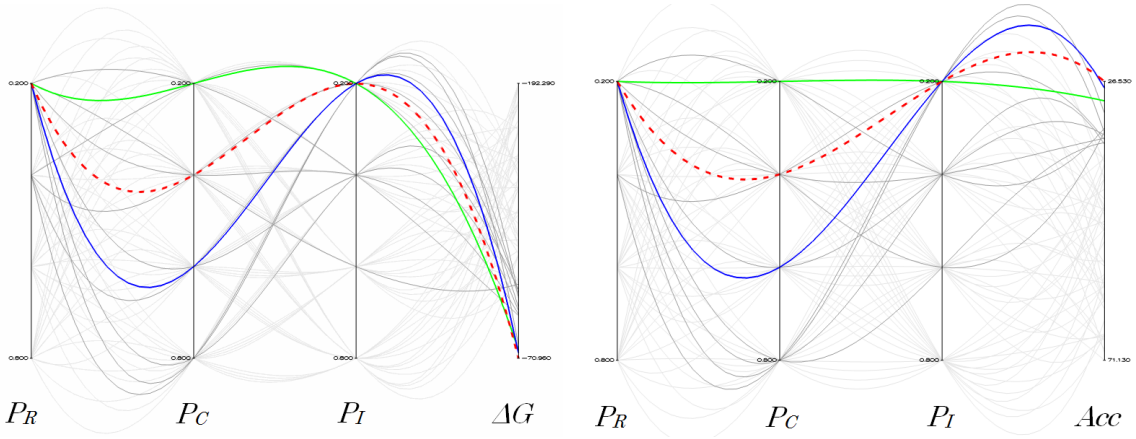


Figure 9.12: Curves mapping to worst 20% fitness values on the left and the worst 20% accuracy values on the right for *X. laevis*. The bottom 5% are shown in colour.

in figure 9.12 are an exact mirror of the best patterns, with common values of $P_R = 0.2$ and $P_I = 0.2$ and different values of P_C. The patterns resulting in the best accuracy also had $P_R = 0.8$ in common (refer to figure 9.11, right hand side) but were less dependant on the value of either P_C or P_I.

Thus a pattern emerges where higher values of P_R and P_I result in better fitness solutions and lower values of P_R and P_I result in worse fitness solutions. The graph shows that the value of P_C is not very important in determining whether the outcome of the

fitness of the swarm will be good or bad. Good and bad fitness results were obtained for any given value of P_C. Remember that P_C determines the influence of the personal best and neighbourhood best on the particle's position. It can't be said that P_C will have no effect in all circumstances. There might be other RNA sequences or other problem domains entirely where P_C may play a bigger role in the fitness that a swarm attains. It can also be concluded that there is a relationship between the two parameters P_R and P_I where the fitness is concerned. Only when both have high values, that is $P_R = 0.8$ and $P_I = 0.8$, do the results show the best fitness, and only when the parameters both have low values does it result in the worst fitness. In this instance the values of P_R and P_I plays a bigger role in the performance of SetPSO. So it can not be concluded that P_C has no effect at all, it only has a lesser effect than P_R and P_I.

Parallel coordinate visualisations for the rest of the sequences are investigated next. The first visualisations represent the best and worst fitness experiments for *D. virilis*. Figure 9.13 illustrates the curves mapping to the best fitnesses and accuracy and figure 9.14 illustrates the curves mapping to the worst fitnesses and accuracy for experiments using *D. virilis*.

The fitness visualisations look almost identical to those of *X. laevis*, with the two parameters P_R and P_I showing the same behaviour as they did previously. Again, $P_R = 0.8$ and $P_I = 0.8$ resulted in the experimental runs with the best fitness while parameter values of $P_R = 0.2$ and $P_I = 0.2$ resulted in the worst fitness performance. Again, the value of the P_I parameter played a lesser role in determining the accuracy of SetPSO. P_C does not seem to influence the performance of the particle swarm optimiser in terms of fitness or accuracy.

The parallel coordinates visualisation for *A. lagunensis* is given in figures 9.15 and 9.16. Again, the same patterns seen for *X. laevis* and *D. virilis* are observed for *A. lagunensis*.

The only visualisations that looked different are those of *S. cerevisiae* and *H. marismortui*. Most of the parameter combinations on these two sequences led to the same

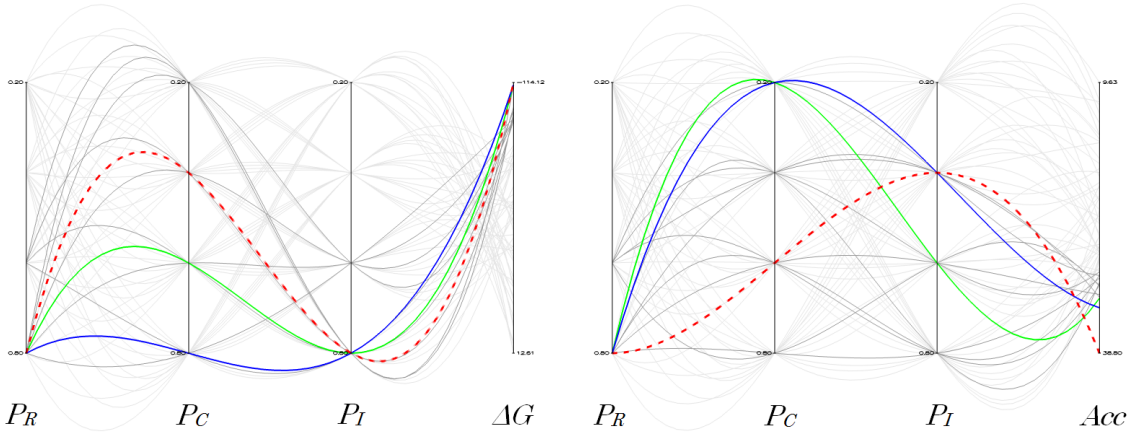


Figure 9.13: Curves mapping to top 20% fitness values on the left and the top 20% accuracy values on the right for *D. virilis*. The top 5% are shown in colour.

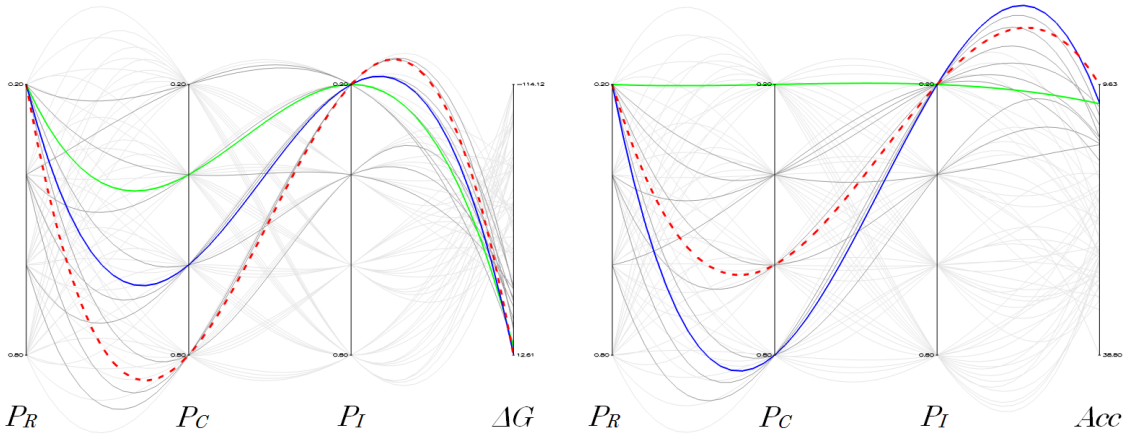


Figure 9.14: Curves mapping to worst 20% fitness values on the left and the worst 20% accuracy values on the right for *D. virilis*. The bottom 5% shown in colour.

optimum fitness, because the search space is so much smaller for these sequences. The parallel coordinates visualisation for *S. cerevisiae* is given in figure 9.17 for interest's sake. All 64 parameter combinations are shown in this figure. Only a handful of curves, the ones with a weak combination of parameter values, do not end up at the optimum fitness. The most patterns have blue curves because said patterns ended up on the same fitness value (the highest fitness). Figure 9.18 illustrates the accuracy of the different parameter values on *S. cerevisiae*. The most patterns are green which corresponds to the

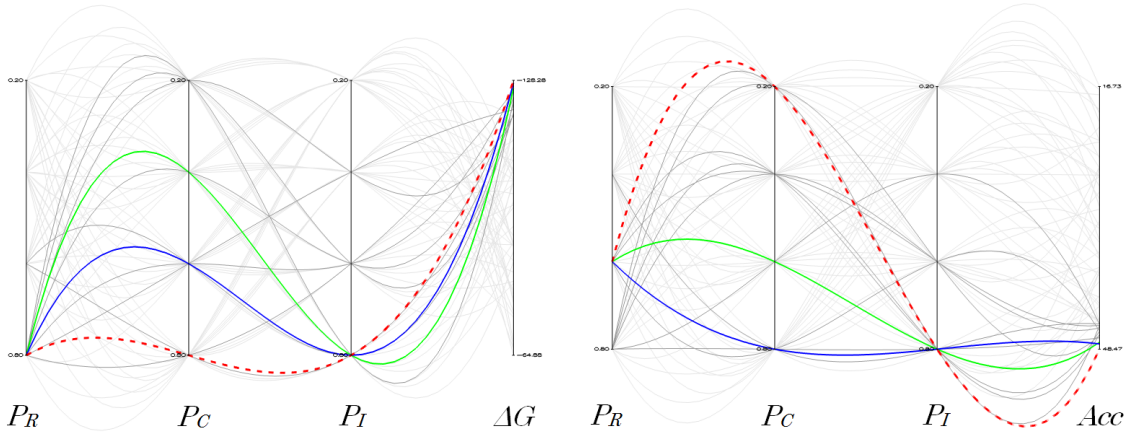


Figure 9.15: Curves mapping to top 20% fitness values on the left and the top 20% accuracy values on the right for *A. lagunensis*. The top 5% are shown in colour.

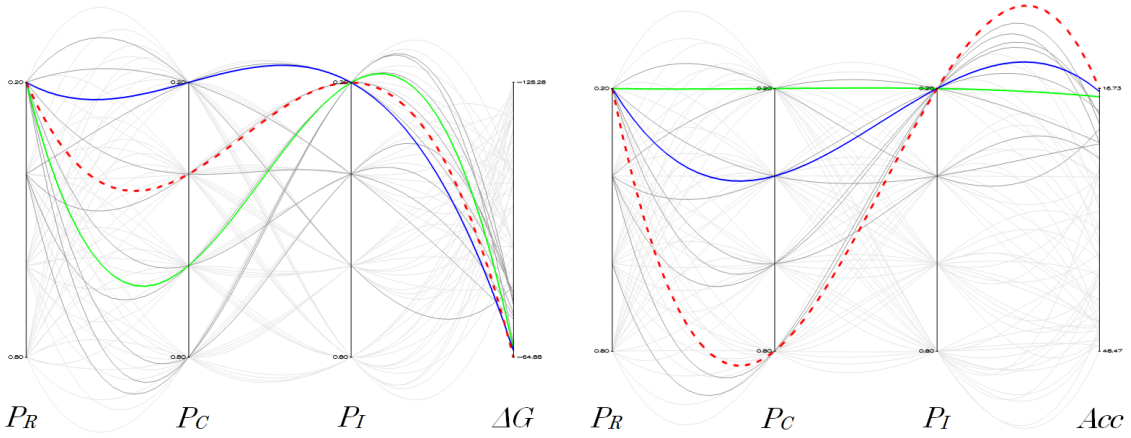


Figure 9.16: Curves mapping to worst 20% fitness values on the left and the top 20% accuracy values on the right for *A. lagunensis*. The bottom 5% are shown in colour.

most fit patterns (blue) in figure 9.17. Some of the patterns with weaker fitness actually have better accuracy (up to 30.17 correct base pairs). The underlying data for figures 9.17 and 9.18 is listed in table A.15. The combinations of parameters resulting in weak fitnesses can be seen in this underlying data. The *H. marismortui* sequence showed a similar result to *S. cerevisiae* and is therefore not visualised here.

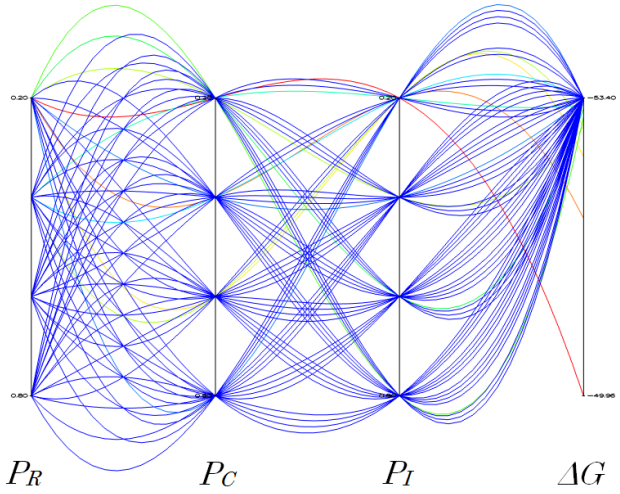


Figure 9.17: Parallel coordinates visualisation for *S. cerevisiae* fitness.

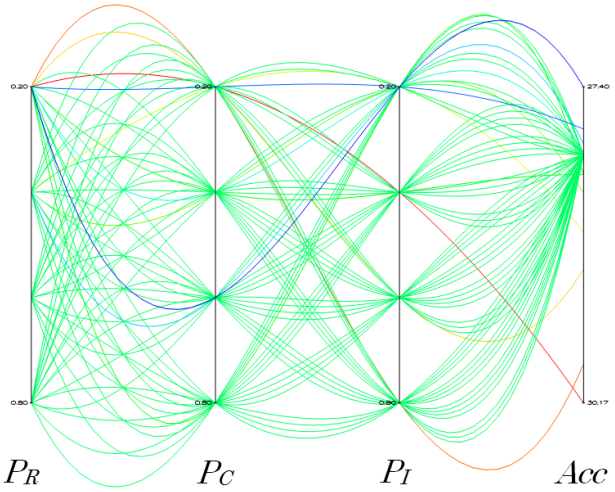


Figure 9.18: Parallel coordinates visualisation for *S. cerevisiae* accuracy.

9.6.2 Accuracy under linear decreasing entropy

Table 9.3 shows the results obtained from the linear decreasing entropy (LDE) experiments compared to a constant entropy (CE) weight, $P_I = 0.8$. The linear decreasing entropy experiments and constant entropy were done with all combinations of P_R and P_C and the appropriate value of P_I. Only the combinations resulting in the best fitness are shown in table 9.3. Only the longer sequences are discussed because the shorter sequences converged to the same conformation in both the constant entropy and the

linear decreasing entropy experiments, and therefore had the same fitness and accuracy.

Table 9.3: Results of linear decreasing entropy weight's effect on fitness and accuracy.

Entropy	Sequence		ΔG (kcal/mol)	Pairs Predicted	Pairs Correct (%)
CE	*X. laevis*	945nt	-192.3±5.0	242.6±6.0	67.1±10.7 (26.1%)
LDE	*X. laevis*	945nt	-187.7±4.2	240.9±6.3	66.7±13.6 (26.0%)
CE	*D. virilis*	784nt	-114.1±3.9	256.7±4.9	31.2±9.0 (13.4%)
LDE	*D. virilis*	784nt	-108.9±3.9	256.6±4.7	29.5±5.7 (12.7%)
CE	*A. lagunensis*	468nt	-128.3±1.8	128.6±3.0	47.1±7.7 (41.7%)
LDE	*A. lagunensis*	468nt	-125.9±1.9	129.0±3.8	46.5±6.9 (41.2%)

For al three sequences, the constant entropy experiments fared better in terms of the fitness result and the accuracy result.

9.7 Investigating the influence of weights on swarm diversity

Swarm diversity is computed using the current position of each particle relative to the *gbest* position. Swarm diversity gives an indication of the dissimilarity of the particles and hence the amount of the search space that the particles cover. Refer to section 5.5 for a discussion on swarm diversity. This section shows the results of the swarm diversity experiments.

9.7.1 Swarm diversity under constant entropy

The swarm diversity for experiments in which the entropy stays constant for the duration of the experiment is discussed first. Only two sequences (*X. laevis* and *S. cerevisiae*) are considered because these sequences are representative of all the sequences with respect to swarm behaviour during experiments.

Two figures are shown with diversity results for the *X. laevis* sequence to highlight the influence of SetPSO parameters on the diversity of the swarm. The parallel coordinate

visualisation in figure 9.19 shows the top 20% parameter combinations that produces the greatest diversity. The top 5% of combinations are shown in colour. Similarly, figure 9.20 highlights the bottom 20% of combinations resulting in the lowest swarm diversity with those under the 5% limit shown in colour. The axes represent the parameters P_R, P_C, P_I, and diversity, D. The greatest diversity is at the bottom on the fourth axis. It is immediately obvious that the combinations with the greatest diversities all have $P_I = 0.2$ in common. Most of the parameter combinations that resulted in lower diversity had $P_I = 0.8$ in common.

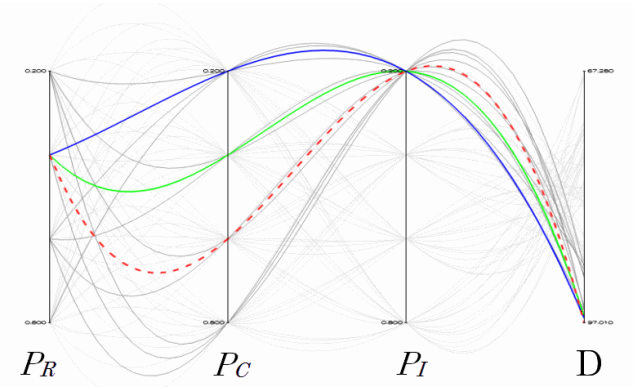


Figure 9.19: Top 20% experiments with highest diversity on *X. laevis* sequence. The axes represent P_R, P_C, P_I and diversity.

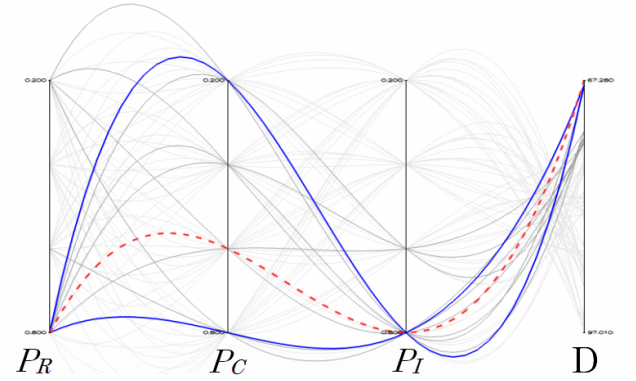


Figure 9.20: Bottom 20% experiments with lowest diversity on *X. laevis* sequence. The axes represent P_R, P_C, P_I and diversity.

When figures 9.19 and 9.20 are examined in conjunction with figures 9.11 and 9.12, which show the parameter combinations for fitness, a pattern emerges where large values

of P_I result in high fitness and low swarm diversity and low values of P_I result in low fitness but higher swarm diversity. The low diversity at higher fitness levels could indicate that the swarm is able to converge on a solution which has a good fitness.

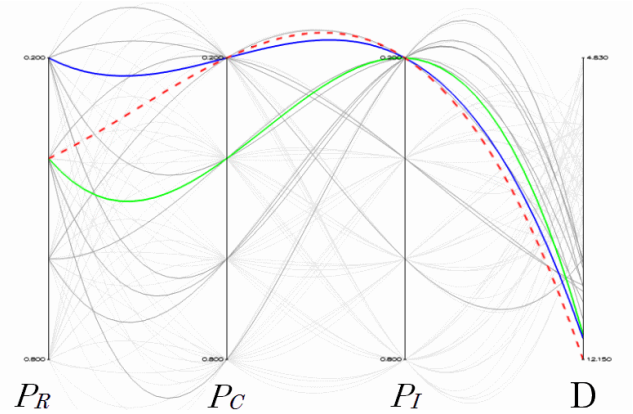


Figure 9.21: Top 20% of experiments with highest diversity on *S. cerevisiae* sequence. The axes represent P_R, P_C, P_I and diversity.

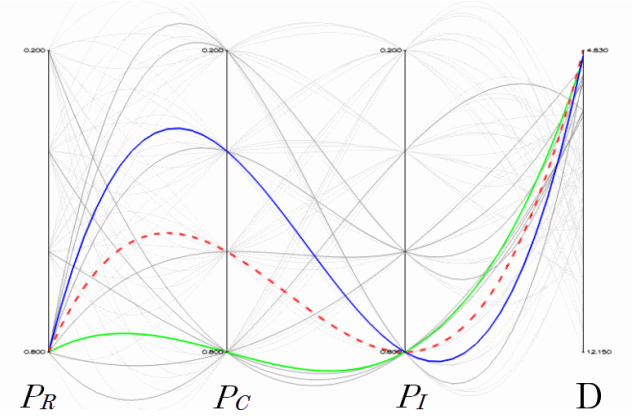


Figure 9.22: Bottom 20% of experiments with lowest diversity on *S. cerevisiae* sequence. The axes represent P_R, P_C, P_I and diversity.

Figures 9.21 and 9.22 show the same tendency for high swarm diversity with smaller values of P_I for the *S. cerevisiae* sequence.

The fact that the swarm diversity is lower with higher entropy values is counter intuitive. But after some investigation, it was concluded that the bias towards adding stems that are in the *pbest* and *nbest* solutions to a particle's velocity dominated the procedure of particle position updates. The larger entropy effectively opened the door

to making the particles more homogeneous by removing many elements from the set and replacing them with stems that are already in use.

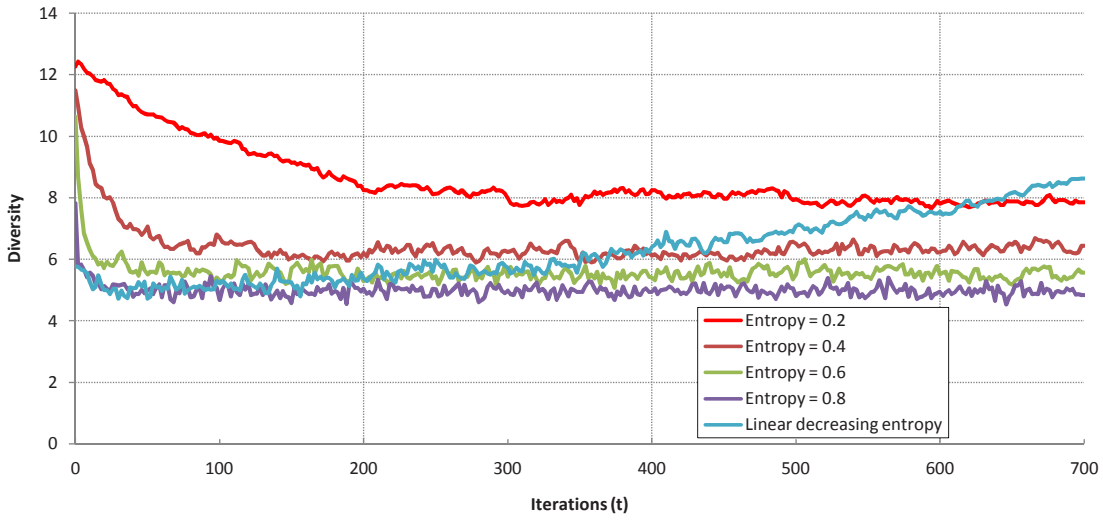
Graph 9.6, which shows the diversity of the swarms over time, illustrates this phenomenon of lower diversity in spite of the higher entropy. The swarm has an initial higher diversity due to the random initialisation process. Gradually, the biased stems dominates and reduces the diversity of the swarm. This process happens at different speeds, depending on the value of P_I. $P_I = 0.2$, for example, does not allow the favoured elements to dominate as quickly and the diversity only levels out at around iteration 300. In contrast, a high value of say $P_I = 0.8$ enables the dominating elements to spread through the swarm fairly quickly in around 20 iterations.

The bias towards elements from the *nbest* and *pbest* positions can be removed, which should result in higher swarm diversity, but that investigation is left as future work.

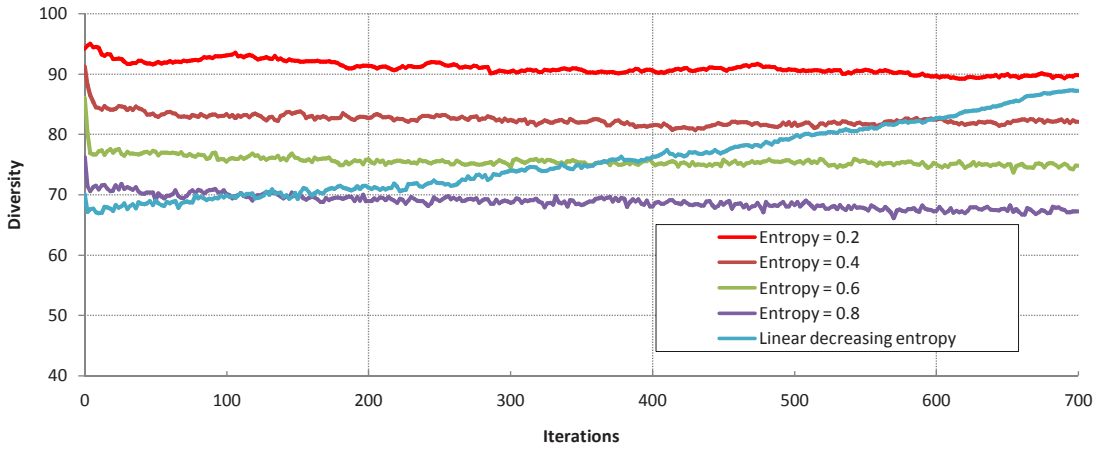
9.7.2 Swarm diversity under linear decreasing entropy

The linear decreasing entropy for each experiment was decreased from the initial value of $P_I = 0.9$ to $P_I = 0.1$ over the 700 iterations. The results of the linear decreasing entropy experiments are shown together with the constant entropy experiments in graphs 9.6 and 9.7 for *S. cerevisiae* and *X. laevis* respectively.

After the analysis of the swarm diversity under various constant entropy values, the results for linear decreasing entropy are more or less in line with expectations: The initial high entropy value results in low swarm diversity, and as the entropy decreases, swarm diversity becomes larger. The complete results for all the linear decreasing entropy experiments can be found in appendix A.



Graph 9.6: The average swarm diversity over the course of an experiment for *S. cerevisiae* sequence.



Graph 9.7: The average swarm diversity over the course of an experiment for *X. laevis* sequence.

9.8 Minimum Stem Length

This section presents the results obtained from experiments done on a universal set containing stems with fewer base pairs (bp), two base pairs to be exact, in addition to the normal stems with three or more base pairs. Section 8.7 gave an overview of the motivation for these experiments.

9.8.1 Reduced minimum stem length

Setting a minimum stem length of three in the previous experiments meant that there are far fewer elements in the universal set of stems for each experiment. The disadvantage is that finer structure elements, where for example an isolated stem of two paired bases exists in a known structure, cannot be modelled. The number of possible stems increases dramatically with the reduction of the minimum stem length to two base pairs. The set with minimum stem lengths of three base pairs is a *subset* of the set containing stems with a minimum stem length of two base pairs. Table 9.4 shows the number of possible stems for each sequence with a minimum stem length of three and two.

Table 9.4: Computational complexity for different minimum stem lengths.

Sequence	Min stem length	Possible stems	Computation time (s)
X. laevis	3 bp	11 329	4 080
X. laevis	2 bp	32 717	-
D. virilis	3 bp	14 460	5 160
D. virilis	2 bp	32 466	-
A. lagunensis	3 bp	2 926	810
A. lagunensis	2 bp	8 351	2 340
H. marismortui	3 bp	220	180
H. marismortui	2 bp	600	240
S. cerevisiae	3 bp	222	190
S. cerevisiae	2 bp	564	210

The table also shows the time taken by SetPSO to complete one experiment consisting

of 30 simulations for each stem length. Allowing stems with a minimum of 2 base pairs results in two to three times more stems and significantly longer computation times, especially for *A. lagunensis*.

Owing to time constraints, only the shorter sequences, namely *A. lagunensis*, *H. marismortui* and *S. cerevisiae* were taken into consideration for testing with a minimum stem length of two base pairs. In order to reduce the number of experiments for this investigation, only experiments with a linear decreasing entropy were done.

9.8.2 Results for reduced stem length SetPSO experiments

The accuracy of the 2 base pair stem experiments is compared with previous experiments and summarised in table 9.5. The parameter combination that resulted in the best fitness (ΔG) is shown for each experiment.

The complete tables with results for the experiments can be found in appendix A, in tables A.8, A.9, A.12, A.13, A.16 and A.17.

Table 9.5: Experimental results and comparison between minimum stem length of 2 and minimum stem length of 3 base pairs.

Sequence	Min stem length	P_R	P_C	ΔG (kcal/mol)	Pairs Predicted	Pairs Correct
A. lagunensis	3 bp	0.8	0.8	-125.9 ± 1.9	129.0 ± 3.8	46.5 ± 6.9
A. lagunensis	2 bp	0.8	0.2	-106.8 ± 3.5	133.3 ± 2.9	44.4 ± 8.7
H. marismortui	3 bp	0.8	0.8	-48.4 ± 0.0	33.0 ± 0.0	16.0 ± 0.0
H. marismortui	2 bp	0.8	0.8	-59.2 ± 0.0	36.0 ± 0.0	31.0 ± 0.0
S. cerevisiae	3 bp	0.8	0.8	-53.4 ± 0.0	40.0 ± 0.0	28.0 ± 0.0
S. cerevisiae	2 bp	0.6	0.6	-54.1 ± 0.0	42.0 ± 0.0	28.2 ± 0.1

A. lagunensis shows a decline

The results given in table 9.5 show very interesting and differing SetPSO behaviour for each sequence tested. The *A. lagunensis* sequence shows an increase in mean free energy (ΔG) when the stem length is reduced to 2 base pairs and a subsequent decrease in

Table 9.6: Average sensitivity of SetPSO, RnaPredict, HelixPSO and *mfold* on *H. marismortui* sequence

			RnaPredict			
Sequence		SetPSO	INN-HB	INN	HelixPSO	*mfold*
H. marismortui	122nt	81.6	42.1	42.1	76.3	76.3

accuracy. The average predicted mean free energy for *A. lagunensis* with a minimum stem length of 3 was -125.9±1.9 $kcal/mol$ but it increased to -106.8±3.5 $kcal/mol$ with a minimum stem length of 2. The resulting accuracy dropped, with on average 2 base pairs less correctly predicted. Remember that the larger stem set for the minimum stem length of 2 is just a superset of the minimum stem length of 3 set, and there is no reason why the same low energy conformations cannot be predicted with the larger set.

The only explanation for the decline in the performance of the algorithm in predicting the *A. lagunensis* sequence is the increased size of the search space. The possible combinations of stems exploded exponentially with the number of stems increasing from 2 926 to 8 351 stems. The actual number of stems in the known conformation containing 2 or fewer base pairs is relatively low compared to the other, longer stems. The known structure is illustrated in figure 9.3. Thus, suboptimal foldings can be predicted with longer stems - relatively accurately - without needing finer grained building blocks to work with. In this case, the additional shorter stems impeded the functioning of the prediction algorithm instead of helping it.

S. cerevisiae remains unchanged

The SetPSO found a marginally better solution for *S. cerevisiae* with respect to the fitness function when adding shorter stems to the set. The mean free energy decreased from -53.4±0.0 $kcal/mol$ to -54.1±0.0 $kcal/mol$, as seen in table 9.5. The number of pairs in the predicted conformation increased from 40 to 42. But ultimately, the accuracy of the predicted structure hardly increased. The accuracy was still around 28 correctly predicted base pairs.

The known structure of *S. cerevisiae* does not contain any stems with less than 3 base pairs. Figure 9.6 shows the known structure for *S. cerevisiae*. Supplementing

the universal stem set U with shorter stems did not improve the accuracy and just adds to the computational complexity of the problem. In the case of *S. cerevisiae*, the thermodynamic model used was inadequate for determining the correct folding.

H. marismortui improves dramatically

The third sequence, *H. marismortui*, showed the most promising results where the additional finer grained building blocks helped to improve the results of SetPSO. The mean free energy was markedly lower from -48.4±0.0 $kcal/mol$ to -59.2±0.0 $kcal/mol$. This only resulted in 3 more predicted base pairs than with longer stems but many more correctly predicted stems. SetPSO could predict a conformation with 31 correct base pairs (sensitivity measure of 81.6%) instead of only 16 correct. Table 9.6 shows the sensitivity obtained by all the algorithms on *H. marismortui* and clearly shows the SetPSO performed the best on this sequence.

The *H. marismortui* sequence was able to benefit greatly from the additional shorter sequences, as the known sequence contains two stems with 2 base pairs. It was thus possible to take this tiny detail into account. For interest's sake, a visual representation of the *H. marismortui* conformations is given. First, figure 9.23 shows the conformation as predicted using stems with a minimum length of 3 base pairs. Figure 9.24 shows a combined structure visualisation of the known structure and the predicted structure of *Haloarcula marismortui* 5S rRNA with shorter stems included. The green bond shows the correctly predicted pairings. The blue bonds are pairings that are in the known structure and not in the predicted structure (false negatives). The red bonds are wrongly predicted pairings (false positives).

Notice in figure 9.24 that both of the stems of length 2 in the known structure are also predicted by SetPSO.

9.9 Conclusion

SetPSO was able to determine structures with relatively low mean free energy values and by implication, very good fitness values according to the objective function. The most accurate structures were mostly not the structures with the best fitness value.

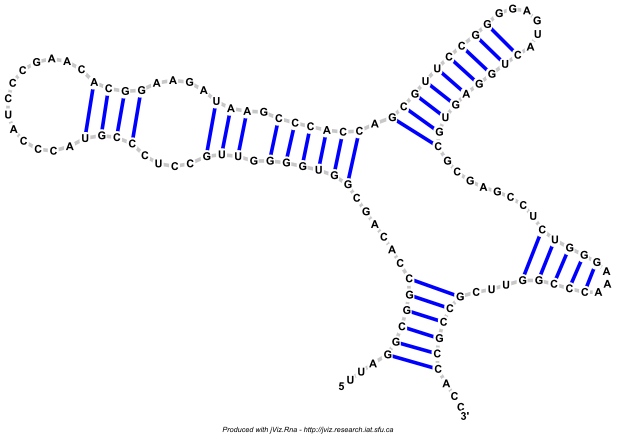


Figure 9.23: The predicted structure of *Haloarcula marismortui* 5S rRNA using stems with a minimum length of 3 base pairs

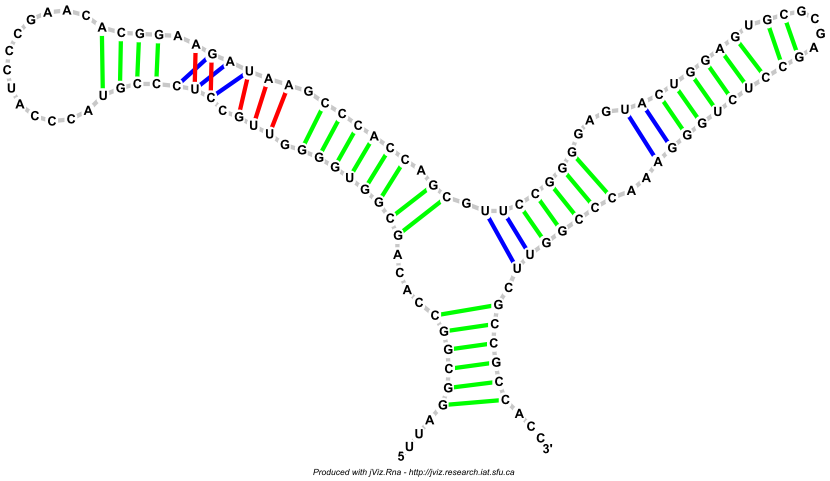


Figure 9.24: The predicted structure of *Haloarcula marismortui* 5S rRNA using stems with a minimum length of 2 base pairs, compared to the known structure

The exact matching criteria used in determining the accuracy score make it difficult to score a good accuracy value. Even though a predicted structure might have significant structural similarities in term of motifs, if the base pairing is off by 1 nucleotide, it scores as a false positive.

The short sequences were easier to predict than the longer sequences. The search space for the shorter sequences was significantly smaller and could easily be searched exhaustively by the swarm. The SetPSO predicted *H. marismortui* solution was not very accurate, in spite of the good fitness value attained for the molecule. This finding was investigated in greater detail in section 9.8.

Although the results showed that the conformations found by the SetPSO are not 100% accurate, obtaining complete accuracy is impossible for a number of reasons. Firstly, an important feature, namely pseudo knots, are not modeled in these experiments because the objective function, the Vienna RNA energy calculation, does not support the inclusion of pseudo knots. Secondly, the thermodynamic energy model is not perfect. The model was built from observations made in laboratory experiments. Finally, the known structures do contain non-canonical base pairs, although in limited numbers. The energy contributions of these non-canonical base pairs cannot be determined by the objective function. The accuracy investigation was carried out in section 9.3.

The predicted structures and the known structures were compared in section 9.4. This can also be seen as an accuracy comparison, but it focused on representing the similarity or dissimilarity of the structures, which cannot easily be deduced from the raw figures alone. Some parts of the conformations were accurately predicted by SetPSO, but if the whole set of sequences is taken into account, the accuracy was not very high.

The results of SetPSO and the *mfold* DPA were compared in section 9.5. The SetPSO managed to obtain results almost on par with *mfold* on some of the sequences.

Section 9.6 aimed to gain insight into the influence that the SetPSO parameters have on the behaviour of the swarm. The fitness and accuracy of the swarms came under the spotlight. In section 9.6 the parameter combinations that resulted in the best fitnesses, highlighted a relationship between the P_R and P_I parameters. The best fitnesses are obtained when both of these parameters have high values ($P_R = 0.8$ and $P_I = 0.8$) and the worst fitnesses are found when these parameters have low values ($P_R = 0.2$ and $P_I = 0.2$). The accuracy of the SetPSO decreased when subject to a linear decreasing entropy. It seems that the bias introduced to favour *nbest* and *pbest* stems drove the swarm to more accurate solutions, even though the swarm exhibited less diversity. The diversity

of the swarms showed counter intuitive results which could, however, be explained by the above-mentioned bias in the SetPSO algorithm in section 9.7.

Fairly diverging results in section 9.8 for the different sequences where observed when reducing the minimum stem length. Fewer stems in the set helped SetPSO to find a better generalised solution with the *A. lagunensis* sequence. The smaller search space aided the SetPSO in finding better solutions.

The predicted structures for *S. cerevisiae* did not differ much between the experiments with longer and shorter stems. More finer grained stems only increased the running time of the algorithm without gaining much in terms of accuracy of the predicted foldings.

However, the *H. marismortui* sequence conformation was much more accurate when the necessary 2 base pair stem building blocks were added. Reducing stem length enabled the algorithm to overcome a major hurdle in predicting a better sequence.

The minimum stem length experiment result raises the following question: What would happen if the minimum stem length was increased for the longer sequences, thereby reducing the complexity of the search space? Of course, this generalisation needs to be balanced by the accuracy of the conformations that can be predicted with only coarser grained building blocks. More research needs to be done to determine the impact of the minimum stem length variable on different sequences.

Chapter 10

Conclusions

This chapter lists the conclusions arrived at by the work presented in this dissertation. These conclusions are elaborated on in section 10.1. In addition, possible future research topics that are suggested by this work are enumerated in section 10.2.

10.1 Conclusions of this dissertation

This study aimed to introduce a new set-based PSO which works on discrete solution spaces. The newly developed SetPSO was used to predict the natural foldings of RNA molecules, given their nucleotide sequences. A background investigation into existing prediction methods showed that the early approaches all used DPA with various thermodynamic energy models. Later techniques used stochastic optimisation algorithms like GAs, coinciding with the algorithms' rise in popularity. Existing stochastic algorithm approaches suggested the idea of using the PSO algorithm to predict RNA conformations.

The new SetPSO algorithm differs from the original PSO which operates in real vector space, by operating on mathematical sets. Each set contains discrete elements which can be added and removed from the set. In the end, the solution provided by SetPSO is a set of elements that has been optimised by the objective function used in the experiments. In order for the algorithm to operate on the set, a new addition operator and subtraction operator were introduced. A distance operator analogous to the string-edit distance operator was introduced to determine the distance from one set

to another in the solution space.

SetPSO also introduced three new parameters, the random add probability P_R, the closing probability P_C and the entropy P_I. Experimental results showed that the P_R and P_I parameters had the most significant influence on the performance as well as the swarm diversity of the SetPSO. Contrary to intuition, the larger the entropy value (P_I), the less diversity the swarm exhibited. This was probably due to a bias introduced in the velocity and position update procedures of the particles which favoured previously found "good" stems. Stems from the *nbest* and *pbest* positions received preference to be inserted into the new position set of a particle in the subsequent iteration. This limits the number of randomly introduced stems. The larger the entropy, the bigger the influence of the stems that had been in other low energy structures.

Eight sequences were tested on SetPSO, namely *Homo sapiens* 16S rRNA (954nt), *Xenopus laevis* mitochondrial 12S rRNA (945 nt), *Drosophila virilis* 16S rRNA (784 nt), *Caenorhabditis elegans* 16S rRNA (697 nt), *Aureoumbra lagunensis* 18S rRNA (468 nt), *Haloarcula marismortui* 5S rRNA (122 nt), *Arthrobacter globiformis* 5S rRNA (123 nt) and *Saccharomyces cerevisiae* 5S rRNA (118 nt). Only five of the sequences were discussed in detail although results for all the sequences are given in appendix A. The conclusions reached for the 5 tested sequences can be generalised to the remaining undiscussed sequences.

The experimental results showed that SetPSO was able to predict structures with relatively low mean free energies as evaluated by the Vienna RNA package. This validates the SetPSO algorithm as an optimisation algorithm that is able to work on discrete mathematical sets and optimise these sets.

Comparisons with RnaPredict, HelixPSO and the *mfold* DPA shows that further improvement needs to be made on SetPSO's ability to predict native conformations. SetPSO compares favourably with RnaPredict but trails HelixPSO and *mfold* in accuracy of the predicted structures. There is an upper bound to the accuracy that SetPSO can achieve. This upper bound is given by *mfold*, because it exhaustively finds the lowest energy structure under the stacking energy model used by both *mfold* and SetPSO. SetPSO can only approach and match the lowest energy structures that *mfold* can obtain. HelixPSO made adaptations to the objective function which improves it's ability to

predict more accurate structures.

Some implementation changes where made to SetPSO which helped in the energy minimisation process. When the stems used as building blocks in the SetPSO were shortened to 2 base pairs, experiments produced mixed results. The *A. lagunensis* sequence saw a reduction in the accuracy because of a reduction in the average fitness of the solutions found when the shorter stems where added to the universal set. The search space became more complex, hence the reduction in average fitness. On the other hand, the *H. marismortui* sequence showed marked improvement in fitness and accuracy. This was because the natural folding for that stem actually contained two stems with only 2 base pairs. This could not previously have been predicted because the minimum stem length was 3 base pairs. This suggests that there is a trade-off between the accuracy of the conformations that can be predicted and the complexity of the search space.

Even with the imperfect energy model and the inability to model pseudo knots in a structure, SetPSO was able to generate reasonable predictions and showed that it might be a useful optimisation algorithm for use in RNA structure prediction.

10.2 Future work

Future work suggested by this study includes the following:

1. Testing the SetPSO discrete optimisation algorithm on different domains and problems. There are a number of discrete optimisation problems that can be used to gauge the effectiveness of SetPSO. These include applications on floorplanning, travelling-sales man problems, packing and knapsack, minimum spanning trees, satisfiability, path optimisation, knights cover problem, n-queens problem, layout optimisation, vehicle routing, urban planning and FPGA placement.

2. Investigating the relation between the accuracy of the predicted stems and the minimum stem length allowed in the structures. Certain heuristics might be deduced to help in predicting better structures on unseen sequences.

3. The tuning of the SetPSO control parameters might result in even better fitness and accuracy results for RNA structure prediction. Further investigation into the

behaviour of SetPSO with varying parameters could also be done.

4. An investigation of the performance and swarm behaviour of SetPSO when the bias towards *pbest* and *nbest* elements is removed from the particle position updates.

Bibliography

[1] V. Ambros. The functions of animal microRNAs. *Nature*, 431:350–355, 2004.

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

[3] G. Benedetti and S. Morosetti. A genetic algorithm to search for optimal and suboptimal rna secondary structures. *Biophys Chem*, 55(3):253–259, August 1995.

[4] M. Birattari and M. Dorigo. How to assess and report the performance of a stochastic algorithm on a benchmark problem: *mean* or *best* result on a number of runs? *Optimization Letters*, 1(1):309–311, 2007.

[5] P.N. Borer, B Dengler, I. Tinoco, and O.C. Uhlenbeck. Stability of RNA hairpin loops: $A_6 - C_m - U_6$. *J. Mol. Biol*, 73:483–496, 1973.

[6] P.N. Borer, B. Dengler, I. Tinoco, O.C. Uhlenbeck, M.D. Levine, D.M. Crothers, and J. Gralla. Improved estimation of secondary structure in ribonucleic acids. *Nature new Biol.*, 246:40–41, 1973.

[7] 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.

[8] Mandelbrot A. Bozarth R.F., Wood H.A. The penicillium stoloniferum virus complex: two similar double-stranded RNA virus-like particles in a single cell. *Virology*, 45(2):516–523, Aug 1971.

[9] R. Brits, A.P. Engelbrecht, and F. van den Bergh. A niching particle swarm optimizer. In *Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning*, pages 692–696, 2002.

[10] Cannone J J, Subramanian S, Schnare M N, Collett J R, D'Souza L M, Du Y, Feng B, Lin N, Madabusi L V, Muller K M, Pande N, Shang Z, Yu N, and Gutell RR. The comparative RNA Web (CRW) site: An online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BioMed Central Bioinformatics*, 3:15, 2002.

[11] M Chastain and I Tinoco Jr. Structural elements in RNA. *Prog Nucleic Acid Res Mol Biol*, 41:131–177, 1991.

[12] Jih-H. Chen, Shu-Yun Le, and Jacob V. Maizel. Prediction of common secondary structures of RNAs: a genetic algorithm approach. *Nucl. Acids Res.*, 28(4):991–999, 2000.

[13] 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.

[14] CIRG. CIlib: Computational Intelligence Library, 2006. University of Pretoria Computational Intelligence Research Group. Available online: [`http://cilib.sourceforge.net`] (Accessed: 20 August 2006).

[15] M Clerc. Discrete particle swarm optimization illustrated by the traveling salesman problem. Technical report, [`http://clerc.maurice.free.fr/pso/`], 2000.

[16] Arthur H. Compton. A quantum theory of the scattering of x-rays by light elements. *Phys. Rev.*, 21(5):483–502, May 1923.

[17] B. Davis, M. Afshar, G. Varani, A. I. Murchie, J. Karn, G. Lentzen, M. Drysdale, J. Bower, A. J. Potter, I. D. Starkey, T. Swarbrick, and F. Aboul-ela. Rational design of inhibitors of hiv-1 tar rna through the stabilisation of electrostatic "hot spots". *J Mol Biol*, 336(2):343–356, February 2004.

[18] Carlos A. Del Carpio M., Mohamed Ismael, Eichiro Ichiishi, Michihisa Koyama, Momoji Kubo, and Akira Miyamoto. An evolving automaton for RNA secondary structure prediction. In Gary G. Yen, Lipo Wang, Piero Bonissone, and Simon M. Lucas, editors, *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 4533–4540, Vancouver, 6-21 July 2006. IEEE Press.

[19] Alain Deschenes. *A Genetic Algorithm for RNA Secondary Structure Prediction Using Stacking Energy Models*. Masters thesis, Simon Fraser University, Burnaby, BC, Canada, June 2005.

[20] K.J. Doshi, J.J. Cannone, C.W. Cobaugh, and R.R. Gutell. Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for RNA secondary structure prediction. In *BMC Bioinformatics*, volume 5, page 105, 2004.

[21] J.A Doudna and T.R. Cech. The chemical repertoire of natural ribozymes. In *Nature*, pages 222–228, 2002.

[22] R. C. Eberhart and J Kennedy. *Swarm Intelligence*. Morgan Kaufmann, 2001.

[23] R.C. Eberhart, P.K. Simpson, and R.W. Dobbins. *Computational Intelligence PC Tools*. Academic Press Professional, first edition edition, 1996.

[24] A. P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. Wiley and Sons, 2005.

[25] A.P. Engelbrecht. *Computational Intelligence: An Introduction*. Wiley and Sons, 2002.

[26] A.P. Engelbrecht and F. van den Bergh. Effects of swarm size on cooperative particle swarm optimisers. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 892–899, 2001.

[27] R.K. Far and G. Sczakiel. The activity of siRNA in mammalian cells is related to structural target accessibility: a comparison with antisense oligonucleotides. *Nucleic Acids Res*, 31(15):4417–4424, 2003.

[28] Brice Felden. Rna structure: experimental analysis. *Curr Opin Microbiol*, May 2007.

[29] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, USA, 1966.

[30] Nelis Franken. Fluxviz, 2008. Available online: [`http://sourceforge.net/projects/fluxviz/`].

[31] A.S. Fraser. Simulation of genetic systems by automatic digital computers. 1. introduction. *Aust. J. Biol. Sci*, 10:484–491, 1957.

[32] 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, 1986.

[33] S.M. Freier, B.J. Burger, D. Alkema, T. Neilson, and D.H. Turner. Effects of 3' dangling end stacking on the stability of ggcc and ccgg double helixes. *Biochemistry*, 22(26):61986206, 1983.

[34] B. Fürtig, C. Richter, J. Wöhnert, and H. Schwalbe. Nmr spectroscopy of rna. *Chembiochem*, 4(10):936–962, October 2003.

[35] Giegerich R. Gardner P.P. A comprehensive comparison of comparative RNA structure prediction approaches. *BMC Bioinformatics*, 5:140, 2004.

[36] M. Geis and M. Middendorf. A Particle Swarm Optimizer for Finding Minimum Free Energy RNA Secondary Structures. In *Proceedings of Swarm Intelligence Symposium*, pages 1–8, 2007.

[37] M. Geis and M Middendorf. HelixPSO, 2008. HelixPSO program. Available online: [`http://www.bioinf.uni-leipzig.de/Software/HelixPSO/HelixPSO.tar.gz`] (Accessed: 30 August 2008).

[38] J.W. Gibbs. Graphical methods in the thermodynamics of fluids. *Transactions of the Connecticut Academy*, (II):309–342, 1873.

[39] J.W. Gibbs. On the equilibrium of heterogeneous substances. *Transactions of the Connecticut Academy*, (III):108–248, 1876.

[40] J.W. Gibbs. On the equilibrium of heterogeneous substances. *Transactions of the Connecticut Academy*, (III):343–524, 1878.

[41] M. Grne, J. P. Frste, S. Klumann, V. A. Erdmann, and L. R. Brown. Detection of multiple conformations of the E-domain of 5S rRNA from E. coli in solution and in crystals by NMR spectroscopy. *Nucleic Acids Research*, 24:2592–2596, 1996.

[42] Perrey S.W. Hickson R.E., Simon C. The performance of several multiple-sequence alignment programs in relation to secondary-structure features for an rrna sequence. *Molecular Biology and Evolution*, 17:530–539, 2000.

[43] Sharp P.M. Higgins D.G. Clustal: A package for performing multiple sequence alignment on a microcomputer. *Gene*, 73:237–244, 1988.

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

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

[46] John H. Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975.

[47] Robert W. Holley, Jean Apgar, George A. Everett, James T. Madison, Mark Marquisee, Susan H. Merrill, John Robert Penswick, and Ada Zamir. Structure of a Ribonucleic Acid. *Science*, 147(3664):1462–1465, 1965.

[48] A Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[49] A Inselberg and B Dimsdale. Parallel coordinates: A tool for visualizing multidimensional geometry. In *Visualization 90*, pages 361–378, 1990.

[50] J A Jaeger, D H Turner, and M Zuker. Improved predictions of secondary structures for RNA. In *PNAS*, pages 7706–7710, 1989.

[51] J.A Jaeger, D.H. Turner, and M Zuker. A comparison of optimal and suboptimal RNA secondary structures predicted by free energy minimization with structures determined by phylogenetic comparison. In *Nucleic Acids Research*, pages 2707–2714, 1991.

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

[53] J Kennedy. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1931–1938, 1999.

[54] J Kennedy and R C Eberhart. A Discrete Binary Version of the Particle Swarm Algorithm. In *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, pages 4104–4109, 1997.

[55] J. Kennedy and R.C. Eberhart. Particle Swarm Optimisation. In *Proceedings of the International Conference on Neural Networks*, pages 1942–1948, 1995.

[56] J Kennedy and R Mendes. Population Structure and Particle Performance. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1671–1676, 2002.

[57] C.A. Kidner and R.A. Martienssen. The developmental role of microRNA in plants. *Current Opinion on Plant Biology*, 8:38–44, 200.

[58] R. Kierzek, M.H. Caruthers, C.E. Longfellow, D. Swinton, D.H. Turner, and S.M. . Polymer-supported RNA synthesis and its application to test the nearest neighbor model for duplex stability. *Biochemistry*, 25:7840–7846, 1986.

[59] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*. The MIT Press, December 1992.

[60] M. P. Latham, D. J. Brown, S. A. McCallum, and A. Pardi. Nmr methods for studying the structure and dynamics of rna. *Chembiochem*, 6(9):1492–1505, September 2005.

[61] V. I. Levenshtein. Appeared in english as: Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710, 1966.

[62] D H Mathews, J Sabina, M Zuker, and D H Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol*, 288:911–940, 1999.

[63] 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 *American Chemical Society*, pages 246–257, 1998.

[64] 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. In *Proceedings of the National Academy of Sciences, USA*, volume 101, page 72877292, 2004.

[65] James Mattson. *The Pioneers of NMR and Magnetic Resonance in Medicine: The Story of MRI.* Bar-Ilan University Press, 1996.

[66] McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29, 1990.

[67] S. Naka, T. Genji, T. Yura, and Y. Fukuyama. Practical distribution state estimation using hybrid particle swarm optimization. *IEEE Power Engineering Society Winter Meeting.*, 2:815–820, January 2001.

[68] M. Neethling and A.P. Engelbrecht. Determining RNA secondary structure using set-based particle swarm optimization. In Gary G. Yen, Simon M. Lucas, Gary Fogel, Graham Kendall, Ralf Salomon, Byoung-Tak Zhang, Carlos A. Coello Coello, and Thomas Philip Runarsson, editors, *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 1670–1677, Vancouver, BC, Canada, 16-21 July 2006. IEEE Press.

[69] R. Nussinov and A.B. Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 77, pages 6309–6313, 1980.

[70] R. Nussinov, G. Piecznik, J.R. Grigg, and D.J. Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied Mathematics*, 35:68–82, 1978.

[71] M. Overhoff, M. Alken, R.K. Far, M. Lemaitre, B. Lebleu, G. Sczakiel, and I Robbins. Local (rna) target structure influences siRNA efficacy: A systematic global analysis. *Journal of Molecular Biology*, 348(4):871–881, 2005.

[72] Phillips L.A. Park J.J. Isolation and purification of double-stranded RNA fragments from retrovirus RNA. *Prep Biochem*, 9(3):261–272, 1979.

[73] J.S. Parker, S.M. Roe, and D. Barford. Structural insights into mRNA recognition from a piwi domain-sirna guide complex. *Nature*, 434:663–666, 2005.

[74] E.S. Peer, F. van den Bergh, and A.P. Engelbrecht. Using Neighborhoods with the Guaranteed Convergence PSO. In *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 235–242, 2003.

[75] J.M. Pipas and J.E. McMahon. Method for predicting rna secondary structure. In *Proc Natl Acad Sci U S A*, volume 72, pages 2017–2021, 1975.

[76] A. Ratnaweera, S. Halgamuge, and H. Watson. Particle swarm optimization with self-adaptive acceleration coefficients. In *Proceedings of the First International Conference on Fuzzy Systems and Knowledge Discovery*, pages 264–268, 2003.

[77] Bart Rylander. *Computational complexity and the genetic algorithm.* Doctoral dissertation, University of Idaho, USA, November 2001.

[78] J. SantaLucia Jr. and D. H. Turner. Measuring the thermodynamics of rna secondary structure formation. *Biopolymers*, 44(3):309–319, 1997.

[79] L Schoofs and B Naudts. Swarm Intelligence on the Binary Constraint Satisfaction Problem. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1444–1449, 2002.

[80] S. Schubert, A. Grunweller, V.A. Erdmann, and J. Kurreck. Influences siRNA efficacy: Systematic analysis of intentionally designed binding regions. *Journal of Molecular Biology*, 348(4):883–93, 2005.

[81] M J Serra and D H Turner. Predicting thermodynamic properties of RNA. *Methods in enzymology.*, 259, 1995.

[82] B. A. Shapiro and J Navetta. A massively parallel genetic algorithm for rna secondary structure prediction. *The Journal of Supercomputing*, 8(3):195–207, November 1994.

[83] Bruce A. Shapiro, Jin Chu Wu, David Bengali, and Mark J. Potts. The massively parallel genetic algorithm for RNA folding: MIMD implementation and population variation . *Bioinformatics*, 17(2):137–148, 2001.

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

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

[86] LX Shen, Z Cai, and I Tinoco Jr. RNA structure at high resolution. *FASEB*, 11(9):10231033, August 1995.

[87] Y. Shi and R.C. Eberhart. A Modifed Particle Swarm Optimizer. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 69–73, 1998.

[88] Y. Shi and R.C. Eberhart. A modified particle swarm optimizer. In *Proceedings of the IEEE Congress on Evolutionary Computation,*, pages 69–73, 1998.

[89] Y. Shi and R.C. Eberhart. Particle swarm optimization: Developments, applications and resources. In *Proceedings of the IEEE Congress on Evolutionary Computation,*, volume 1, pages 27–30, 2001.

[90] N. Sugimoto, S.M. Freier, and D.H. Turner. RNA structure prediction. *Annu. Rev. Biophys. Chem.*, 17:167–92, 1998.

[91] N. Sugimoto, R. Kierzek, and D.H. Turner. Sequence dependence for the energetics of dangling ends and terminal mismatches in ribonucleic acid. *Biochemistry*, 26:4554–4558, 1987.

[92] Gibson T.J. Thompson J.D., Higgins D.G. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22:46734680, 1994.

[93] I. Tinoco and O.C. Uhlenbeck. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, 1971.

[94] IC Trelea. The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters*, 85(6):317–325, 2003.

[95] D. Tsou and C. MacNish. Adaptive particle swarm optimisation for high-dimensional highly convex search spaces. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 2, pages 783–789, December 2003.

[96] F. H. van Batenburg, A. P. Gultyaev, and C. W. Pleij. An apl-programmed genetic algorithm for the prediction of rna secondary structure. *J Theor Biol*, 174(3):269–280, June 1995.

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

[98] F. van den Bergh and A.P. Engelbrecht. A study of particle swarm optimization particle trajectories. *Information Sciences*, 2005.

[99] A E Walter, D H Turner, J Kim, M H Lyttle, P Mller, D H Mathews, and M Zuker. Coaxial stacking of helixes enhances binding of oligoribonucleotides and improves predictions of RNA folding. *Proceedings of the National Academy of Sciences of the United States of America.*, 91, 1994.

[100] M. S. Waterman and T. F. Smith. RNA secondary structure: A complete mathematical analysis. *Math. Biosc*, 42:257–266, 1978.

[101] Michael S. Waterman, Dedicated To, and John R. Kinney. Secondary structure of single - stranded nucleic acids. In *Studies on foundations and combinatorics, Advances in mathematics supplementary studies, Academic Press N.Y., 1:167 – 212*, pages 167–212, 1978.

[102] JD Watson and FHC Crick. A structure for deoxyribose nucleic acid. *Nature*, 3(171):737–738, April 25, 1953.

[103] K C Wiese, E Glen, and A Vasudevan. jViz.RNA - A Java Tool for RNA Secondary Structure Visualization. In *IEEE Transactions on NanoBioscience*, pages 212–218, 2005.

[104] Kay C. Wiese, Alain A. Deschnes, and Andrew G. Hendriks. RnaPredictAn evolutionary algorithm for RNA secondary structure prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(1):25–41, 2008.

[105] Kay C. Wiese and Andrew Hendriks. Comparison of P-RnaPredict and mfold - algorithms for RNA secondary structure prediction. *Bioinformatics*, page btl043, 2006.

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

[107] W. David Wilson, Lynda Ratmeyer, Min Zhao, Daoyuan Ding, Adrian W. McConnaughie, Arvind Kumar, and David W. Boykin. Design and analysis of rna structure-specific agents as potential antivirals. *Biochemistry*, 9(2):187–196, 1996.

[108] Pace N Woese C. Probing RNA structure, function, and history by comparative analysis. In *The RNA World*, pages 91–117. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, 1993.

[109] M Wu, J A McDowell, and D H Turner. A periodic table of symmetric tandem mismatches in RNA. *Biochemistry.*, 34, 1995.

[110] T. Xia, J. SantaLucia Jr., M.E. Burkard, R. Kierzek, S.J. Schroeder, X. Jiao, C. Cox, and D.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.

[111] H. Yoshida, Y. Fukuyama, S. Takayama, and Y Nakanishi. A particle swarm optimization for reactive power and voltage control in electric power systems considering voltage security assessment. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, volume 6, pages 497–502, October 1999.

[112] Taotao Zhang, Maozu Guo, and Quan Zou. Rna secondary structure prediction based on forest representation and genetic algorithm. In *ICNC '07: Proceedings of the Third International Conference on Natural Computation (ICNC 2007)*, pages 370–374, Washington, DC, USA, 2007. IEEE Computer Society.

[113] M Zuker. On finding all suboptimal foldings of an RNA molecule. In *Science*, pages 48–52, 1989.

[114] M Zuker. Prediction of RNA secondary structure by energy minimization. In *Computer Analysis of Sequence Data*, pages 267–294, 1994.

[115] M Zuker. Mfold web server for nucleic acid folding and hybridization prediction. In *Nucleic Acids Research*, pages 3406–3415, 2003.

[116] M Zuker, D.H. Mathews, and D.H. Turner. Algorithms and thermodynamics for RNA secondary structure prediction: a practical guide. In *RNA Biochemistry and Biotechnology*, 1999.

[117] M Zuker and P Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. In *Nucleic Acids Research*, pages 133–148, 1981.

Appendix A

Complete Results for All Sequences

This appendix gives the results obtained for the 5 sequences discussed in the main body and 3 additional sequences and the comparative results obtained from *mfold*.

A.1 *Xenopus laevis* mitochondrial 12S rRNA

A.1.1 SetPSO results, constant entropy

Table A.1: Experimental results for *Xenopus laevis* mitochondrial 12S rRNA showing the mean and standard deviation over 30 samples.

P_R	P_C	P_I	ΔG **kcal/mol**	**Swarm diversity**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
0.8	0.6	0.8	-192.3 ± 5.0	67.3 ± 3.6	242.6 ± 6.0	67.1 ± 10.7	26.1
0.8	0.8	0.8	-192.2 ± 4.0	67.8 ± 3.3	242.3 ± 5.8	67.0 ± 10.1	26.1
0.8	0.4	0.8	-191.9 ± 3.3	68.1 ± 2.8	242.0 ± 6.1	67.2 ± 8.9	26.1
0.8	0.2	0.8	-190.8 ± 3.9	67.8 ± 2.6	239.7 ± 5.4	64.7 ± 8.4	25.2
0.8	0.2	0.6	-185.1 ± 4.2	74.2 ± 2.6	242.4 ± 5.8	67.3 ± 11.0	26.2
0.8	0.6	0.6	-184.3 ± 3.8	74.8 ± 3.0	241.4 ± 7.0	71.1 ± 11.7	27.7
0.8	0.8	0.6	-183.0 ± 5.5	75.4 ± 2.9	242.2 ± 6.1	67.9 ± 7.8	26.4
0.8	0.4	0.6	-182.7 ± 3.9	74.7 ± 3.0	241.5 ± 8.0	67.2 ± 12.1	26.1
0.8	0.2	0.4	-176.1 ± 5.0	80.7 ± 2.3	242.5 ± 5.6	67.9 ± 11.4	26.4
0.8	0.4	0.4	-173.8 ± 4.3	81.2 ± 3.1	244.2 ± 6.5	69.4 ± 13.0	27.0
0.6	0.2	0.8	-172.9 ± 3.8	76.1 ± 1.7	230.6 ± 6.3	57.7 ± 14.0	22.4
0.8	0.6	0.4	-172.9 ± 6.3	82.1 ± 3.3	242.7 ± 6.3	62.9 ± 8.9	24.5
0.6	0.4	0.8	-172.0 ± 5.2	76.8 ± 2.0	234.1 ± 5.9	60.3 ± 12.5	23.5
0.8	0.8	0.4	-171.9 ± 4.8	81.7 ± 2.5	243.4 ± 6.4	69.9 ± 12.3	27.2
0.6	0.8	0.8	-170.8 ± 5.6	74.6 ± 2.7	231.0 ± 6.6	59.3 ± 11.7	23.1
0.6	0.6	0.8	-170.7 ± 3.7	75.3 ± 2.5	231.1 ± 6.6	61.5 ± 15.4	23.9
0.6	0.2	0.6	-161.6 ± 5.3	80.3 ± 2.4	228.2 ± 4.5	59.0 ± 14.7	22.9
0.6	0.8	0.6	-160.5 ± 6.1	81.1 ± 2.0	230.5 ± 5.1	58.8 ± 10.6	22.9
0.6	0.6	0.6	-159.3 ± 6.2	81.9 ± 3.0	229.8 ± 6.1	58.7 ± 13.2	22.9
0.8	0.2	0.2	-158.8 ± 6.8	90.2 ± 2.8	245.8 ± 5.6	61.7 ± 10.9	24.0
0.6	0.4	0.6	-158.2 ± 4.6	81.9 ± 2.4	229.6 ± 7.5	57.4 ± 9.5	22.3

Continued on next page

Table A.1 – continued from previous page

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.8	0.4	0.2	-157.2 ± 9.1	89.8 ± 3.2	246.9 ± 6.9	61.9 ± 15.1	24.1
0.8	0.6	0.2	-153.2 ± 7.7	89.8 ± 3.1	246.3 ± 4.8	61.9 ± 11.0	24.1
0.4	0.2	0.8	-151.2 ± 7.4	77.7 ± 2.3	214.4 ± 7.6	50.5 ± 11.0	19.7
0.8	0.8	0.2	-150.6 ± 6.7	89.5 ± 2.9	247.6 ± 6.3	59.7 ± 11.9	23.2
0.6	0.2	0.4	-149.4 ± 4.9	87.4 ± 2.6	229.5 ± 5.5	55.2 ± 12.3	21.5
0.4	0.8	0.8	-146.7 ± 6.7	77.5 ± 2.5	217.8 ± 5.8	50.1 ± 9.0	19.5
0.4	0.4	0.8	-146.4 ± 5.5	77.6 ± 2.4	212.9 ± 7.8	46.6 ± 11.1	18.1
0.6	0.4	0.4	-146.3 ± 6.3	87.8 ± 2.3	228.5 ± 6.3	56.1 ± 11.7	21.8
0.6	0.6	0.4	-146.1 ± 6.8	86.9 ± 1.9	231.7 ± 6.5	55.6 ± 12.9	21.6
0.4	0.6	0.8	-146.0 ± 6.0	77.9 ± 2.2	215.6 ± 6.7	52.1 ± 12.8	20.3
0.6	0.8	0.4	-143.7 ± 8.5	86.7 ± 2.4	232.5 ± 8.2	50.3 ± 7.7	19.6
0.4	0.2	0.6	-137.1 ± 5.4	82.4 ± 2.2	213.7 ± 8.5	46.8 ± 11.8	18.2
0.4	0.6	0.6	-135.6 ± 4.8	83.0 ± 2.4	216.7 ± 7.5	48.9 ± 10.4	19.0
0.4	0.8	0.6	-134.8 ± 6.2	82.6 ± 1.3	216.4 ± 6.9	49.5 ± 10.0	19.2
0.4	0.4	0.6	-134.4 ± 6.0	83.0 ± 2.1	215.9 ± 6.9	46.5 ± 8.8	18.1
0.6	0.2	0.2	-130.2 ± 8.0	95.6 ± 2.9	235.6 ± 6.5	50.5 ± 10.9	19.6
0.6	0.6	0.2	-125.2 ± 7.5	95.1 ± 2.4	233.5 ± 6.2	47.3 ± 9.0	18.4
0.6	0.4	0.2	-124.4 ± 7.7	96.1 ± 3.1	233.3 ± 6.6	47.9 ± 9.1	18.6
0.4	0.2	0.4	-123.3 ± 7.6	87.9 ± 1.5	213.5 ± 7.3	43.8 ± 10.9	17.0
0.4	0.8	0.4	-121.0 ± 7.5	88.7 ± 2.0	218.6 ± 7.9	46.6 ± 10.6	18.1
0.4	0.4	0.4	-120.2 ± 7.1	88.1 ± 2.4	216.3 ± 8.1	47.0 ± 9.9	18.3
0.6	0.8	0.2	-120.2 ± 5.5	95.4 ± 2.4	236.2 ± 8.1	42.2 ± 10.9	16.4
0.2	0.8	0.8	-118.5 ± 6.7	73.3 ± 2.6	194.4 ± 8.7	42.1 ± 11.3	16.4
0.4	0.6	0.4	-117.0 ± 7.3	88.6 ± 2.4	215.9 ± 7.9	45.4 ± 9.9	17.7
0.2	0.4	0.8	-116.8 ± 4.5	73.3 ± 2.7	188.9 ± 7.4	42.0 ± 10.2	16.3
0.2	0.2	0.8	-116.4 ± 4.9	73.2 ± 1.8	186.2 ± 6.9	39.6 ± 7.7	15.4
0.2	0.6	0.8	-115.5 ± 5.8	74.0 ± 2.2	190.5 ± 8.4	39.9 ± 9.5	15.5

Continued on next page

Table A.1 – continued from previous page

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.2	0.8	0.6	-105.4 ± 5.6	78.5 ± 2.3	193.0 ± 9.1	36.3 ± 9.0	14.1
0.2	0.6	0.6	-104.8 ± 6.8	78.1 ± 2.4	191.0 ± 9.3	38.8 ± 10.8	15.1
0.2	0.2	0.6	-104.8 ± 6.3	77.1 ± 1.8	185.5 ± 8.9	38.1 ± 8.9	14.8
0.2	0.4	0.6	-103.6 ± 6.3	77.3 ± 1.9	185.0 ± 8.9	34.6 ± 11.5	13.5
0.4	0.2	0.2	-101.3 ± 6.0	96.5 ± 2.4	218.1 ± 8.3	38.8 ± 12.5	15.1
0.4	0.4	0.2	-100.0 ± 7.7	96.2 ± 1.7	218.7 ± 5.2	35.8 ± 9.2	13.9
0.4	0.8	0.2	-99.3 ± 6.0	95.5 ± 2.7	223.4 ± 7.8	34.7 ± 11.7	13.5
0.4	0.6	0.2	-97.8 ± 7.3	97.0 ± 2.0	224.4 ± 8.8	39.5 ± 10.0	15.4
0.2	0.8	0.4	-93.5 ± 7.1	83.2 ± 2.2	198.1 ± 9.7	36.6 ± 12.5	14.2
0.2	0.6	0.4	-91.9 ± 6.2	83.5 ± 2.0	195.4 ± 7.8	35.1 ± 8.0	13.7
0.2	0.4	0.4	-90.4 ± 4.5	82.7 ± 2.5	190.8 ± 10.6	35.6 ± 10.6	13.9
0.2	0.2	0.4	-89.8 ± 5.5	81.6 ± 2.0	185.8 ± 7.2	34.0 ± 9.0	13.2
0.2	0.8	0.2	-76.7 ± 5.3	90.5 ± 2.2	208.8 ± 6.8	31.6 ± 8.5	12.3
0.2	0.2	0.2	-74.6 ± 5.7	90.8 ± 2.3	193.5 ± 8.9	29.6 ± 10.2	11.5
0.2	0.6	0.2	-73.3 ± 5.3	91.7 ± 2.4	200.6 ± 6.4	27.6 ± 8.0	10.7
0.2	0.4	0.2	-71.0 ± 5.4	91.4 ± 2.5	195.5 ± 7.0	26.5 ± 9.0	10.3

A.1.2 SetPSO results, linear decreasing entropy

Table A.2: Experimental results for *Xenopus laevis* mitochondrial 12S rRNA showing the mean and standard deviation over 30 samples.

		ΔG	**Swarm**	**Pairs**	**Pairs**	**%**
P_R	P_C	**kcal/mol**	**diversity**	**Predicted**	**Correct**	**Correct**
0.2	0.2	-111.2 ± 5.8	87.4 ± 2.0	184.7 ± 9.2	34.6 ± 8.5	13.5
0.2	0.4	-106.2 ± 4.9	88.1 ± 1.5	183.4 ± 7.2	32.3 ± 7.8	12.6
0.2	0.6	-108.0 ± 7.5	88.3 ± 1.6	188.3 ± 8.3	34.3 ± 12.8	13.4
0.2	0.8	-111.2 ± 7.0	87.5 ± 2.2	191.7 ± 9.6	37.3 ± 9.3	14.5
0.4	0.2	-142.8 ± 4.9	93.0 ± 2.1	210.9 ± 7.7	47.6 ± 12.5	18.5
0.4	0.4	-140.0 ± 6.4	93.8 ± 2.1	211.9 ± 8.5	46.0 ± 10.9	17.9
0.4	0.6	-139.8 ± 5.7	94.2 ± 2.7	216.3 ± 8.0	47.2 ± 11.4	18.4
0.4	0.8	-139.1 ± 6.3	93.2 ± 2.4	214.7 ± 8.2	45.5 ± 13.5	17.7
0.6	0.2	-165.6 ± 5.0	93.7 ± 2.2	231.6 ± 6.8	58.9 ± 10.6	22.9
0.6	0.4	-163.9 ± 5.1	93.7 ± 2.0	228.4 ± 7.3	58.3 ± 11.0	22.7
0.6	0.6	-165.8 ± 5.4	91.9 ± 2.5	228.3 ± 7.2	62.4 ± 11.7	24.3
0.6	0.8	-161.6 ± 6.8	93.5 ± 2.2	229.5 ± 8.5	55.7 ± 12.7	21.7
0.8	0.2	-187.7 ± 4.2	87.7 ± 1.8	240.9 ± 6.3	66.7 ± 13.6	26.0
0.8	0.4	-186.4 ± 4.4	87.2 ± 2.7	240.7 ± 6.3	66.9 ± 10.9	26.0
0.8	0.6	-183.9 ± 5.9	87.2 ± 2.3	238.9 ± 7.0	62.4 ± 9.0	24.3
0.8	0.8	-182.9 ± 6.0	87.4 ± 2.6	241.0 ± 5.1	62.2 ± 10.8	24.2

A.1.3 *mfold* results

Table A.3: *Mfold* results for *Xenopus laevis* mitochondrial 12S rRNA.

mfold ΔG **kcal/mol**	***efn2*** ΔG **kcal/mol**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
-250.6	-222.85	249	92	35.8
-249.6	-219.75	251	71	27.6
-248.8	-219.63	241	97	37.7
-248.6	-218.69	246	84	32.7
-248	-216.51	245	**113**	**44.0**
-248	-213.01	242	100	38.9
-247.8	-210.87	241	84	32.7
-247.4	-209.26	243	74	28.8
-247.2	-218.3	246	79	30.7
-247.1	-215.7	244	76	29.6
-246.7	-211.01	238	69	26.8
-246.5	-221.02	244	88	34.2
-246.5	-214.62	245	68	26.5
-246.3	-223.07	248	101	39.3
-245.3	-214.07	250	103	40.1
-245	-217.38	248	62	24.1
-244.7	-215.17	243	80	31.1
-244.3	-223.49	246	86	33.5
-243.7	-213.42	237	73	28.4
-243.6	-205.9	242	91	35.4
-242.5	-202.27	251	81	31.5

A.2 *Drosophila virilis* 16S rRNA

A.2.1 SetPSO results, constant entropy

Table A.4: Experimental results for *Drosophila virilis* 16S rRNA showing the mean and standard deviation over 30 samples.

P_R	P_C	P_I	ΔG **kcal/mol**	**Swarm diversity**	**Pairs Predicted**	**Pairs Correct**	**Percentage Correct**
0.8	0.8	0.8	-113.03±4.10	57.86±3.13	255.10±5.14	30.70±6.69	12.0
0.8	0.6	0.8	-112.90±3.87	57.21±4.05	256.57±5.46	31.47±9.24	12.3
0.8	0.2	0.8	-112.66±1.73	56.01±2.44	255.47±4.68	28.43±8.19	11.1
0.8	0.2	0.6	-102.42±4.65	66.72±2.24	255.13±7.23	32.87±10.97	12.9
0.8	0.4	0.6	-100.78±4.96	68.58±2.53	253.87±5.12	29.63±11.09	11.7
0.6	0.2	0.8	-100.05±4.08	65.73±1.99	244.83±7.22	28.30±9.03	11.6
0.6	0.4	0.8	-98.10±4.93	66.85±2.80	244.07±6.20	28.73±9.46	11.8
0.8	0.8	0.6	-97.16±7.10	70.88±2.83	252.63±6.10	28.23±10.21	11.2
0.6	0.8	0.8	-96.87±6.60	67.88±2.26	247.50±7.88	27.30±7.66	11.0
0.8	0.6	0.6	-96.53±4.64	71.01±3.20	252.47±6.29	32.47±9.78	12.9
0.6	0.6	0.8	-96.10±4.48	68.76±2.59	247.37±6.46	30.40±10.20	12.3
0.8	0.2	0.4	-84.65±5.30	75.77±2.21	249.50±5.21	33.90±7.31	13.6
0.6	0.2	0.6	-83.62±3.92	74.74±1.64	244.43±6.61	28.00±12.90	11.5
0.4	0.2	0.8	-81.17±4.18	69.68±2.08	231.07±7.35	28.87±13.85	12.5
0.8	0.4	0.4	-80.04±5.83	78.43±2.12	248.97±6.33	31.47±8.76	12.6
0.6	0.4	0.6	-79.96±6.72	74.82±1.95	242.53±6.76	30.03±11.09	12.4
0.4	0.8	0.8	-77.19±7.46	71.57±2.60	232.53±8.53	27.80±9.39	12.0
0.4	0.4	0.8	-76.73±4.45	70.84±1.67	230.50±7.20	27.93±9.18	12.1
0.4	0.6	0.8	-76.33±5.71	71.41±1.81	232.10±7.68	27.33±10.20	11.8
0.6	0.6	0.6	-75.94±5.36	75.36±1.73	239.33±5.55	27.07±9.48	11.3
0.8	0.6	0.4	-75.72±5.38	79.50±1.39	248.03±7.23	38.80±8.17	15.6

Continued on next page

Table A.4 – continued from previous page

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	Percentage Correct
0.8	0.8	0.4	-74.16±5.86	80.90±2.01	248.20±5.75	28.03±9.25	11.3
0.6	0.8	0.6	-73.59±5.32	77.29±1.93	239.33±6.91	31.57±10.07	13.2
0.4	0.2	0.6	-65.47±5.88	74.95±2.04	229.47±8.92	26.20±10.09	11.4
0.6	0.2	0.4	-61.77±5.51	81.54±1.89	237.00±8.06	30.43±10.29	12.8
0.4	0.4	0.6	-59.05±5.03	77.21±1.87	226.53±7.47	29.67±11.17	13.1
0.2	0.2	0.8	-59.00±3.89	67.64±1.73	206.00±7.95	23.37±9.34	11.3
0.6	0.4	0.4	-57.90±7.80	83.17±2.73	236.17±9.22	32.77±11.82	13.9
0.4	0.6	0.6	-57.20±4.95	77.64±2.12	225.67±8.61	29.90±13.25	13.2
0.8	0.2	0.2	-55.49±5.27	88.36±2.30	245.93±7.74	24.87±9.13	10.1
0.4	0.8	0.6	-55.00±6.95	78.59±2.18	226.30±8.67	27.90±13.57	12.3
0.2	0.4	0.8	-54.59±4.68	68.45±2.15	203.53±7.75	26.13±10.20	12.8
0.6	0.6	0.4	-54.29±6.60	84.65±2.27	235.80±7.14	27.60±11.77	11.7
0.6	0.8	0.4	-52.17±5.32	84.45±1.98	235.77±7.53	25.93±12.24	11.0
0.2	0.6	0.8	-51.16±4.81	69.26±2.04	201.97±9.06	20.73±11.30	10.3
0.2	0.8	0.8	-50.80±4.64	68.48±1.87	205.07±6.44	21.83±10.92	10.6
0.8	0.4	0.2	-49.32±5.62	90.33±2.12	244.77±6.85	27.93±10.83	11.4
0.8	0.6	0.2	-47.51±6.56	90.69±2.46	242.70±6.30	26.27±10.47	10.8
0.8	0.8	0.2	-44.79±8.96	91.81±1.83	242.90±4.93	24.77±9.02	10.2
0.4	0.2	0.4	-42.85±5.58	82.52±1.66	223.30±6.47	20.40±10.50	9.1
0.2	0.2	0.6	-41.42±4.46	71.61±1.94	199.30±7.52	22.03±10.87	11.1
0.2	0.4	0.6	-37.58±6.17	72.88±2.18	199.30±8.60	23.97±11.31	12.0
0.4	0.4	0.4	-37.01±4.92	83.56±1.47	223.23±7.54	24.07±10.55	10.8
0.2	0.6	0.6	-35.37±5.24	73.75±1.76	199.97±7.50	20.97±11.00	10.5
0.4	0.6	0.4	-34.95±5.44	83.93±1.56	217.87±5.48	22.33±11.67	10.2
0.2	0.8	0.6	-33.87±6.14	73.94±1.66	200.30±8.96	19.33±9.35	9.7
0.4	0.8	0.4	-31.35±7.53	85.09±1.38	219.73±8.03	22.37±10.41	10.2
0.6	0.2	0.2	-30.49±6.53	92.05±2.12	232.63±8.04	23.33±13.18	10.0

Continued on next page

Table A.4 – continued from previous page

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	Percentage Correct
0.6	0.4	0.2	-22.89±5.60	93.77±1.52	231.57±8.31	17.83±6.77	7.7
0.2	0.2	0.4	-21.28±6.14	78.31±2.13	197.40±8.35	18.90±8.60	9.6
0.6	0.6	0.2	-19.83±5.94	94.15±1.92	229.03±7.72	19.77±7.83	8.6
0.2	0.4	0.4	-18.39±4.68	78.77±1.98	195.80±6.96	16.13±8.22	8.2
0.6	0.8	0.2	-18.16±6.67	95.38±1.50	232.37±6.07	15.30±8.81	6.6
0.2	0.6	0.4	-14.12±3.57	79.94±1.96	198.10±7.05	16.00±9.55	8.1
0.2	0.8	0.4	-12.58±4.70	81.22±1.31	199.63±7.09	15.73±9.92	7.9
0.4	0.2	0.2	-10.08±6.16	92.06±1.85	216.57±8.75	18.37±10.99	8.5
0.4	0.4	0.2	-4.20±4.45	93.52±1.88	216.33±6.23	15.33±8.30	7.1
0.4	0.6	0.2	-1.22±4.65	94.44±2.09	214.73±7.31	14.00±8.15	6.5
0.4	0.8	0.2	0.44±6.50	94.98±2.07	217.20±8.27	12.97±8.19	6.0
0.2	0.2	0.2	6.99±6.61	87.67±1.49	197.50±8.57	11.70±6.78	5.9
0.2	0.4	0.2	8.59±7.12	87.74±1.58	194.93±7.84	12.03±9.32	6.2
0.2	0.6	0.2	11.98±6.36	88.73±1.93	196.20±6.35	9.63±7.38	4.9
0.2	0.8	0.2	12.61±5.31	88.89±2.18	196.97±6.70	11.60±7.59	5.9

A.2.2 SetPSO results, linear decreasing entropy

Table A.5: Experimental results for *Drosophila virilis* 16S rRNA showing the mean and standard deviation over 30 samples.

		ΔG	**Swarm**	**Pairs**	**Pairs**	**%**
P_R	P_C	**kcal/mol**	**diversity**	**Predicted**	**Correct**	**Correct**
0.2	0.2	-53.8 ± 4.8	82.8 ± 1.3	203.8 ± 5.5	20.8 ± 10.4	8.9
0.2	0.4	-51.0 ± 6.6	83.0 ± 2.0	202.0 ± 7.6	19.4 ± 9.9	8.3
0.2	0.6	-47.7 ± 5.1	83.3 ± 1.3	201.1 ± 9.3	19.9 ± 11.7	8.5
0.2	0.8	-49.2 ± 7.4	83.8 ± 2.4	202.1 ± 8.3	20.6 ± 9.4	8.9
0.4	0.2	-78.8 ± 4.8	87.8 ± 1.4	229.7 ± 6.3	26.3 ± 7.8	11.3
0.4	0.4	-73.8 ± 6.3	88.0 ± 1.7	229.5 ± 8.5	27.0 ± 10.4	11.6
0.4	0.6	-72.9 ± 5.0	87.5 ± 1.7	226.8 ± 5.5	23.4 ± 11.3	10.1
0.4	0.8	-71.4 ± 7.4	88.5 ± 1.6	228.9 ± 6.6	27.4 ± 6.9	11.8
0.6	0.2	-95.9 ± 4.4	88.3 ± 1.3	247.3 ± 7.5	29.5 ± 10.0	12.6
0.6	0.4	-95.0 ± 3.8	88.4 ± 1.3	244.5 ± 7.5	32.2 ± 9.3	13.8
0.6	0.6	-92.6 ± 4.9	89.6 ± 2.0	245.7 ± 5.9	25.9 ± 10.3	11.1
0.6	0.8	-92.5 ± 5.0	89.0 ± 1.3	245.0 ± 7.3	26.7 ± 9.9	11.5
0.8	0.2	-108.9 ± 3.9	82.8 ± 2.1	256.6 ± 4.7	29.5 ± 5.7	12.7
0.8	0.4	-107.7 ± 3.6	84.0 ± 1.9	255.6 ± 5.8	25.1 ± 6.9	10.8
0.8	0.6	-108.3 ± 3.7	84.6 ± 1.9	256.4 ± 4.3	27.9 ± 8.1	12.0
0.8	0.8	-107.6 ± 5.4	84.2 ± 2.3	254.3 ± 5.7	33.0 ± 8.1	14.2

A.2.3 *mfold* results

Table A.6: *mfold* results for suboptimal foldings of *Drosophila virilis* 16S rRNA

mfold ΔG **kcal/mol**	***efn2*** ΔG **kcal/mol**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
-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

Continued on next page

Table A.6 – continued from previous page

mfold ΔG **kcal/mol**	***efn2*** ΔG **kcal/mol**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
-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

A.3 *Aureoumbra lagunensis* 18S rRNA

A.3.1 SetPSO results, constant entropy

Table A.7: Experimental results for *Aureoumbra lagunensis* 18S rRNA showing the mean and standard deviation over 30 samples.

P_R	P_C	P_I	ΔG **kcal/mol**	**Swarm diversity**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
0.8	0.8	0.8	-128.3 ± 1.8	29.1 ± 2.2	128.6 ± 3.0	47.1 ± 7.7	41.7
0.8	0.6	0.8	-127.7 ± 0.9	29.1 ± 2.3	128.0 ± 2.7	44.7 ± 6.1	39.6
0.8	0.4	0.8	-126.5 ± 1.9	31.5 ± 2.1	127.1 ± 2.8	46.7 ± 4.8	41.3
0.8	0.6	0.6	-126.1 ± 2.8	33.2 ± 2.0	128.0 ± 3.5	45.0 ± 5.8	39.8
0.8	0.8	0.6	-125.8 ± 2.3	32.0 ± 2.7	127.7 ± 4.1	43.3 ± 5.0	38.3
0.8	0.4	0.6	-124.5 ± 2.1	34.6 ± 2.5	126.8 ± 3.8	45.9 ± 7.3	40.6
0.8	0.2	0.8	-124.0 ± 0.9	31.8 ± 2.7	127.1 ± 2.5	45.6 ± 6.2	40.4
0.8	0.2	0.6	-122.9 ± 1.5	34.6 ± 1.4	127.9 ± 2.0	46.8 ± 8.2	41.4
0.8	0.6	0.4	-121.5 ± 3.1	39.8 ± 2.2	127.2 ± 3.0	45.5 ± 8.9	40.2
0.6	0.6	0.8	-120.8 ± 3.6	36.2 ± 1.8	122.9 ± 3.6	47.6 ± 9.8	42.2
0.6	0.8	0.8	-120.5 ± 3.1	35.4 ± 2.2	122.9 ± 5.2	47.8 ± 8.6	42.3
0.8	0.2	0.4	-120.2 ± 2.3	40.0 ± 2.3	127.8 ± 3.5	45.2 ± 8.2	40.0
0.8	0.4	0.4	-120.0 ± 3.4	40.3 ± 2.1	127.5 ± 2.3	43.0 ± 7.3	38.0
0.8	0.8	0.4	-119.8 ± 4.1	39.6 ± 2.5	127.2 ± 3.8	45.1 ± 7.2	39.9
0.6	0.4	0.8	-117.7 ± 2.6	36.9 ± 1.6	122.0 ± 4.0	46.3 ± 8.1	41.0
0.6	0.2	0.8	-117.5 ± 3.1	37.8 ± 1.2	120.3 ± 5.8	48.5 ± 12.0	42.9
0.6	0.6	0.6	-114.8 ± 3.3	40.3 ± 1.5	122.3 ± 4.6	45.0 ± 11.0	39.8
0.6	0.8	0.6	-114.4 ± 3.5	39.9 ± 1.0	122.1 ± 3.2	46.3 ± 7.7	40.9
0.6	0.4	0.6	-113.0 ± 3.2	41.2 ± 1.5	121.0 ± 3.4	45.6 ± 8.4	40.4
0.6	0.2	0.6	-111.7 ± 3.4	40.8 ± 1.3	120.0 ± 4.3	42.0 ± 9.1	37.2
0.4	0.8	0.8	-110.2 ± 4.3	37.9 ± 1.7	118.4 ± 4.8	45.4 ± 9.5	40.2

Continued on next page

Table A.7 – continued from previous page

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.8	0.2	0.2	-109.8 ± 4.7	47.0 ± 1.3	125.4 ± 3.1	35.6 ± 11.8	31.5
0.4	0.6	0.8	-108.0 ± 2.8	38.4 ± 1.5	114.8 ± 5.5	43.3 ± 7.6	38.3
0.8	0.4	0.2	-107.9 ± 4.5	46.6 ± 1.5	125.3 ± 4.4	33.6 ± 10.8	29.8
0.6	0.8	0.4	-107.5 ± 5.3	44.6 ± 1.6	122.2 ± 4.0	37.4 ± 8.0	33.1
0.8	0.8	0.2	-106.5 ± 5.5	47.4 ± 1.6	126.7 ± 3.8	31.3 ± 14.6	27.7
0.4	0.2	0.8	-105.8 ± 3.5	39.6 ± 1.5	111.6 ± 5.7	39.0 ± 10.3	34.5
0.8	0.6	0.2	-105.5 ± 6.4	47.8 ± 1.8	125.7 ± 3.7	35.3 ± 13.7	31.2
0.6	0.6	0.4	-105.3 ± 5.0	45.6 ± 1.9	122.8 ± 5.2	37.0 ± 11.6	32.7
0.6	0.4	0.4	-105.2 ± 6.4	45.5 ± 1.8	121.1 ± 5.1	39.7 ± 10.5	35.1
0.4	0.4	0.8	-104.7 ± 3.6	39.7 ± 1.5	110.0 ± 5.7	39.1 ± 10.2	34.6
0.6	0.2	0.4	-104.4 ± 3.3	45.3 ± 1.0	118.1 ± 3.9	42.2 ± 8.5	37.3
0.4	0.8	0.6	-102.1 ± 4.8	41.4 ± 1.5	114.1 ± 4.6	37.5 ± 12.0	33.2
0.4	0.2	0.6	-100.3 ± 4.5	42.2 ± 1.4	111.0 ± 4.7	40.3 ± 8.0	35.6
0.4	0.6	0.6	-99.6 ± 4.1	42.6 ± 1.1	112.1 ± 3.7	39.2 ± 12.2	34.7
0.4	0.4	0.6	-99.0 ± 2.9	42.4 ± 1.4	111.6 ± 4.7	36.7 ± 13.5	32.5
0.2	0.8	0.8	-94.6 ± 4.4	36.6 ± 1.5	106.8 ± 5.6	38.6 ± 12.1	34.1
0.6	0.2	0.2	-92.5 ± 3.7	49.8 ± 1.8	119.2 ± 4.0	27.6 ± 11.3	24.4
0.6	0.4	0.2	-90.9 ± 5.7	50.9 ± 1.3	119.8 ± 4.9	30.3 ± 12.5	26.8
0.2	0.6	0.8	-90.9 ± 3.7	37.3 ± 1.8	101.5 ± 6.3	33.0 ± 10.1	29.2
0.4	0.8	0.4	-90.8 ± 3.6	45.4 ± 1.8	113.7 ± 5.5	28.8 ± 12.6	25.5
0.6	0.6	0.2	-90.7 ± 6.3	50.1 ± 1.5	120.6 ± 4.8	25.4 ± 10.4	22.5
0.4	0.2	0.4	-90.4 ± 4.2	44.8 ± 1.5	109.3 ± 5.5	27.6 ± 10.8	24.4
0.4	0.4	0.4	-90.1 ± 3.7	46.4 ± 0.5	111.6 ± 5.5	27.4 ± 11.1	24.3
0.6	0.8	0.2	-90.0 ± 6.3	50.8 ± 1.6	123.4 ± 3.2	23.3 ± 10.6	20.6
0.4	0.6	0.4	-89.9 ± 5.3	45.9 ± 1.5	112.5 ± 5.0	27.5 ± 10.7	24.3
0.2	0.4	0.8	-88.8 ± 3.7	38.5 ± 2.0	100.9 ± 5.1	38.6 ± 9.6	34.2
0.2	0.8	0.6	-86.7 ± 5.8	39.2 ± 1.7	103.2 ± 5.3	33.9 ± 10.4	30.0

Continued on next page

Table A.7 – continued from previous page

P_R	P_C	P_I	ΔG **kcal/mol**	**Swarm diversity**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
0.2	0.2	0.8	-86.5 ± 3.0	38.1 ± 1.2	97.3 ± 4.3	26.5 ± 8.8	23.5
0.2	0.6	0.6	-84.1 ± 3.7	39.9 ± 1.0	101.3 ± 4.1	31.6 ± 10.5	27.9
0.2	0.4	0.6	-81.9 ± 4.4	40.4 ± 0.9	99.5 ± 5.3	29.5 ± 9.2	26.1
0.2	0.2	0.6	-81.4 ± 3.8	39.3 ± 1.3	96.9 ± 5.9	26.4 ± 11.2	23.4
0.4	0.2	0.2	-79.8 ± 4.5	50.6 ± 1.3	113.2 ± 4.6	23.3 ± 11.4	20.6
0.2	0.8	0.4	-78.8 ± 4.1	42.5 ± 1.2	104.8 ± 4.9	26.6 ± 10.8	23.5
0.4	0.4	0.2	-78.0 ± 4.1	50.7 ± 1.1	114.9 ± 4.5	19.5 ± 8.6	17.3
0.4	0.6	0.2	-77.6 ± 4.3	50.5 ± 1.5	114.7 ± 5.2	19.8 ± 7.2	17.5
0.4	0.8	0.2	-76.9 ± 4.8	49.6 ± 1.2	115.3 ± 4.5	21.0 ± 9.6	18.6
0.2	0.6	0.4	-75.0 ± 4.2	42.8 ± 0.9	101.0 ± 4.9	25.2 ± 10.9	22.3
0.2	0.4	0.4	-74.3 ± 3.6	43.1 ± 0.9	100.4 ± 5.2	23.4 ± 7.7	20.7
0.2	0.2	0.4	-73.7 ± 3.0	42.7 ± 1.7	98.9 ± 6.1	21.6 ± 6.3	19.1
0.2	0.8	0.2	-68.0 ± 4.2	47.5 ± 1.4	109.0 ± 4.7	16.7 ± 7.4	14.8
0.2	0.6	0.2	-66.8 ± 3.2	47.7 ± 1.2	106.7 ± 6.6	19.9 ± 9.2	17.6
0.2	0.2	0.2	-66.4 ± 4.8	47.4 ± 1.2	103.1 ± 5.1	17.7 ± 8.6	15.7
0.2	0.4	0.2	-64.9 ± 2.8	47.5 ± 1.0	103.5 ± 4.4	17.1 ± 7.2	15.1

A.3.2 SetPSO results, linear decreasing entropy

Table A.8: Experimental results for *Aureoumbra lagunensis* 18S rRNA showing the mean and standard deviation over 30 samples.

		ΔG	**Swarm**	**Pairs**	**Pairs**	**%**
P_R	P_C	**kcal/mol**	**diversity**	**Predicted**	**Correct**	**Correct**
0.8	0.8	-125.9 ± 1.9	40.9 ± 1.9	129.0 ± 3.8	46.5 ± 6.9	41.1
0.8	0.6	-124.6 ± 1.6	43.0 ± 1.0	127.7 ± 2.7	44.2 ± 7.0	39.1
0.8	0.4	-123.7 ± 1.7	44.5 ± 1.9	127.7 ± 2.9	44.7 ± 6.5	39.5
0.8	0.2	-123.0 ± 2.1	45.0 ± 1.3	128.5 ± 3.8	45.3 ± 6.6	40.1
0.6	0.6	-115.2 ± 3.6	48.4 ± 1.2	119.6 ± 4.5	44.6 ± 9.8	39.5
0.6	0.8	-114.9 ± 3.9	48.0 ± 1.4	122.1 ± 3.7	43.1 ± 9.8	38.2
0.6	0.4	-114.3 ± 3.0	48.7 ± 1.3	121.1 ± 4.7	45.7 ± 9.2	40.4
0.6	0.2	-113.9 ± 3.6	49.0 ± 1.3	118.9 ± 4.4	44.3 ± 8.1	39.2
0.4	0.2	-102.5 ± 3.8	49.3 ± 0.9	109.5 ± 5.2	37.1 ± 11.2	32.8
0.4	0.6	-102.2 ± 4.3	49.3 ± 1.7	113.7 ± 5.5	41.2 ± 9.8	36.5
0.4	0.8	-102.0 ± 3.7	48.6 ± 1.2	113.4 ± 6.0	39.4 ± 10.0	34.9
0.4	0.4	-99.8 ± 3.2	49.5 ± 1.1	111.6 ± 3.8	40.9 ± 8.5	36.2
0.2	0.8	-89.5 ± 5.1	45.5 ± 1.1	102.9 ± 6.5	30.5 ± 10.6	27.0
0.2	0.6	-85.3 ± 4.5	46.4 ± 1.5	101.5 ± 5.2	26.4 ± 10.6	23.3
0.2	0.2	-83.2 ± 4.0	46.3 ± 1.4	98.3 ± 6.3	27.1 ± 9.4	24.0
0.2	0.4	-82.4 ± 3.5	46.1 ± 1.0	98.6 ± 5.6	24.8 ± 9.6	22.0

A.3.3 SetPSO results, minimum stem length of 2

Table A.9: Experimental results for *Aureoumbra lagunensis* 18S rRNA with a minimum stem length of 2 showing the mean and standard deviation over 30 samples.

		ΔG	**Swarm**	**Pairs**	**Pairs**	**%**
P_R	P_C	**kcal/mol**	**diversity**	**Predicted**	**Correct**	**Correct**
0.8	0.2	-106.8 ± 3.5	67.4 ± 1.2	133.3 ± 2.9	44.4 ± 8.7	39.3
0.8	0.4	-106.3 ± 4.9	68.9 ± 1.2	134.0 ± 3.4	43.5 ± 9.1	38.5
0.8	0.8	-105.7 ± 5.4	68.6 ± 2.1	132.1 ± 4.1	37.9 ± 12.6	33.5
0.8	0.6	-103.9 ± 5.0	68.4 ± 1.2	130.8 ± 3.3	40.0 ± 13.8	35.4
0.6	0.2	-96.6 ± 6.4	70.9 ± 1.8	127.2 ± 4.7	37.0 ± 12.0	32.7
0.6	0.4	-92.3 ± 4.6	71.0 ± 1.6	126.3 ± 4.0	34.5 ± 9.2	30.5
0.6	0.8	-92.1 ± 4.0	72.1 ± 1.4	126.6 ± 3.4	34.7 ± 11.9	30.7
0.6	0.6	-91.2 ± 6.7	71.7 ± 1.1	126.1 ± 3.9	32.5 ± 11.9	28.8
0.4	0.2	-82.8 ± 4.7	70.2 ± 1.4	119.7 ± 4.2	27.7 ± 9.8	24.5
0.4	0.4	-80.3 ± 6.3	71.4 ± 1.5	120.1 ± 5.0	30.4 ± 10.4	26.9
0.4	0.8	-76.3 ± 5.3	71.9 ± 1.4	118.1 ± 3.7	29.1 ± 10.3	25.8
0.4	0.6	-76.3 ± 4.9	71.9 ± 1.6	119.2 ± 4.2	27.2 ± 10.9	24.1
0.2	0.2	-62.3 ± 5.3	66.6 ± 2.4	105.9 ± 5.7	19.7 ± 8.8	17.5
0.2	0.4	-58.6 ± 5.2	67.0 ± 1.7	106.2 ± 3.9	20.3 ± 8.9	18.0
0.2	0.6	-58.6 ± 7.2	66.9 ± 1.0	105.8 ± 6.2	19.2 ± 8.7	17.0
0.2	0.8	-56.7 ± 5.8	67.1 ± 0.9	106.6 ± 4.1	20.7 ± 7.8	18.3

A.3.4 *mfold* results

Table A.10: *mfold* results for suboptimal foldings of *Aureoumbra lagunensis*

mfold ΔG **kcal/mol**	***efn2*** ΔG **kcal/mol**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
-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.4 *Haloarcula marismortui* 5S rRNA

A.4.1 SetPSO results, constant entropy

Table A.11: Experimental results for *Haloarcula marismortui* 5S rRNA showing the mean and standard deviation over 30 samples.

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.2	0.4	0.6	-48.4 ± 0.0	8.4 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.2	0.4	0.8	-48.4 ± 0.0	7.7 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.2	0.6	0.6	-48.4 ± 0.0	7.5 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.2	0.6	0.8	-48.4 ± 0.0	7.0 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.2	0.8	0.6	-48.4 ± 0.0	6.6 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.2	0.8	0.8	-48.4 ± 0.0	6.1 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.2	0.4	-48.4 ± 0.0	9.9 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.2	0.6	-48.4 ± 0.0	8.8 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.2	0.8	-48.4 ± 0.0	8.2 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.4	0.4	-48.4 ± 0.0	9.4 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.4	0.6	-48.4 ± 0.0	8.0 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.4	0.8	-48.4 ± 0.0	7.5 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.6	0.6	-48.4 ± 0.0	7.5 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.6	0.8	-48.4 ± 0.0	7.0 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.8	0.4	-48.4 ± 0.0	8.4 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.8	0.6	-48.4 ± 0.0	7.2 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.8	0.8	-48.4 ± 0.0	6.6 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.2	0.4	-48.4 ± 0.0	9.0 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.2	0.6	-48.4 ± 0.0	7.9 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.2	0.8	-48.4 ± 0.0	7.4 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.4	0.4	-48.4 ± 0.0	8.9 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1

Continued on next page

Table A.11 – continued from previous page

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.6	0.4	0.6	-48.4 ± 0.0	7.2 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.4	0.8	-48.4 ± 0.0	6.9 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.6	0.4	-48.4 ± 0.0	8.2 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.6	0.6	-48.4 ± 0.0	7.2 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.6	0.8	-48.4 ± 0.0	6.2 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.8	0.4	-48.4 ± 0.0	7.9 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.8	0.6	-48.4 ± 0.0	7.0 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.8	0.8	-48.4 ± 0.0	6.3 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.2	0.2	-48.4 ± 0.0	9.7 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.2	0.4	-48.4 ± 0.0	7.6 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.2	0.6	-48.4 ± 0.0	7.1 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.2	0.8	-48.4 ± 0.0	6.7 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.4	0.2	-48.4 ± 0.0	10.0 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.4	0.4	-48.4 ± 0.0	7.4 ± 0.6	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.4	0.6	-48.4 ± 0.0	6.5 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.4	0.8	-48.4 ± 0.0	6.1 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.6	0.4	-48.4 ± 0.0	7.2 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.6	0.6	-48.4 ± 0.0	6.2 ± 0.1	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.6	0.8	-48.4 ± 0.0	5.6 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.8	0.4	-48.4 ± 0.0	6.6 ± 0.7	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.8	0.6	-48.4 ± 0.0	5.8 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.8	0.8	-48.4 ± 0.0	5.6 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.2	0.4	0.4	-48.3 ± 0.0	9.4 ± 0.0	32.8 ± 0.0	16.7 ± 2.6	44.0
0.2	0.6	0.4	-48.3 ± 0.0	8.7 ± 0.0	33.0 ± 0.0	15.9 ± 0.0	41.8
0.8	0.6	0.2	-48.3 ± 0.0	9.6 ± 0.8	33.0 ± 0.0	15.9 ± 0.0	41.8
0.4	0.6	0.4	-48.2 ± 0.0	8.9 ± 0.0	32.9 ± 0.0	15.7 ± 0.0	41.4
0.6	0.2	0.2	-48.2 ± 0.0	11.2 ± 0.0	32.8 ± 0.0	16.6 ± 2.7	43.7

Continued on next page

Table A.11 – continued from previous page

P_R	P_C	P_I	ΔG **kcal/mol**	**Swarm diversity**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
0.2	0.2	0.8	-48.2 ± 0.0	8.7 ± 0.0	32.8 ± 0.0	16.1 ± 2.0	42.4
0.2	0.8	0.4	-48.1 ± 0.4	7.8 ± 0.5	33.2 ± 0.0	15.0 ± 2.8	39.5
0.6	0.4	0.2	-48.0 ± 0.0	11.3 ± 0.0	33.0 ± 0.0	15.5 ± 0.9	40.7
0.8	0.8	0.2	-48.0 ± 0.5	9.9 ± 0.4	33.1 ± 0.0	15.1 ± 2.4	39.7
0.2	0.2	0.6	-47.9 ± 0.0	9.2 ± 0.0	32.6 ± 0.0	16.6 ± 3.6	43.6
0.6	0.6	0.2	-47.5 ± 1.0	11.2 ± 0.0	33.0 ± 0.0	14.7 ± 3.7	38.7
0.2	0.2	0.4	-47.2 ± 0.7	10.2 ± 0.0	32.2 ± 0.7	16.9 ± 5.3	44.4
0.4	0.2	0.2	-47.0 ± 1.0	11.6 ± 0.0	32.4 ± 1.2	16.5 ± 5.7	43.5
0.6	0.8	0.2	-47.0 ± 1.2	10.8 ± 0.0	33.1 ± 0.2	13.5 ± 3.0	35.6
0.4	0.4	0.2	-46.6 ± 1.0	11.3 ± 0.0	32.8 ± 1.0	13.8 ± 3.5	36.4
0.4	0.6	0.2	-46.6 ± 1.2	11.0 ± 0.0	33.2 ± 1.0	13.0 ± 4.4	34.3
0.4	0.8	0.2	-46.0 ± 1.1	10.0 ± 1.2	33.6 ± 1.2	10.9 ± 5.9	28.8
0.2	0.6	0.2	-45.5 ± 1.0	10.9 ± 0.0	33.3 ± 1.4	11.5 ± 5.5	30.2
0.2	0.4	0.2	-45.3 ± 1.2	11.2 ± 0.0	33.1 ± 1.4	12.0 ± 4.8	31.6
0.2	0.8	0.2	-45.2 ± 1.4	9.5 ± 1.2	33.6 ± 1.2	9.2 ± 4.0	24.2
0.2	0.2	0.2	-44.5 ± 1.0	11.0 ± 0.0	32.1 ± 1.5	14.4 ± 5.5	37.9

A.4.2 SetPSO results, linear decreasing entropy

Table A.12: Experimental results for *Haloarcula marismortui* 5S rRNA showing the mean and standard deviation over 30 samples.

		ΔG	**Swarm**	**Pairs**	**Pairs**	**%**
P_R	P_C	**kcal/mol**	**diversity**	**Predicted**	**Correct**	**Correct**
0.8	0.8	-48.4 ± 0.0	10.0 ± 0.8	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.6	-48.4 ± 0.0	10.1 ± 0.7	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.4	-48.4 ± 0.0	10.3 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.8	0.2	-48.4 ± 0.0	10.0 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.8	-48.4 ± 0.0	11.0 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.6	-48.4 ± 0.0	11.6 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.4	-48.4 ± 0.0	11.3 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.6	0.2	-48.4 ± 0.0	11.6 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.8	-48.4 ± 0.0	10.9 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.6	-48.4 ± 0.0	11.3 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.4	-48.4 ± 0.0	11.7 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.4	0.2	-48.4 ± 0.0	12.0 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.2	0.8	-48.4 ± 0.0	9.4 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.2	0.6	-48.4 ± 0.0	10.0 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.2	0.4	-48.4 ± 0.0	11.1 ± 0.0	33.0 ± 0.0	16.0 ± 0.0	42.1
0.2	0.2	-47.8 ± 0.5	11.5 ± 0.0	32.6 ± 0.0	15.9 ± 3.2	41.9

A.4.3 SetPSO results, minimum stem length of 2

Table A.13: Experimental results for *Haloarcula marismortui* 5S rRNA with a minimum stem length of 2 showing the mean and standard deviation over 30 samples.

		ΔG	**Swarm**	**Pairs**	**Pairs**	**%**
P_R	P_C	**kcal/mol**	**diversity**	**Predicted**	**Correct**	**Correct**
0.8	0.8	-59.2 ± 0.0	14.7 ± 0.5	36.0 ± 0.0	31.0 ± 0.0	81.6
0.8	0.6	-59.2 ± 0.0	15.9 ± 0.8	36.0 ± 0.0	31.0 ± 0.0	81.6
0.8	0.4	-59.2 ± 0.0	16.6 ± 0.4	36.0 ± 0.0	31.0 ± 0.0	81.6
0.8	0.2	-59.2 ± 0.0	17.3 ± 0.0	36.0 ± 0.0	31.0 ± 0.0	81.6
0.6	0.6	-59.2 ± 0.0	17.4 ± 0.0	36.0 ± 0.0	31.0 ± 0.0	81.6
0.6	0.4	-59.2 ± 0.0	18.1 ± 0.0	36.0 ± 0.0	31.0 ± 0.0	81.6
0.4	0.8	-59.2 ± 0.0	17.0 ± 0.0	36.0 ± 0.0	31.0 ± 0.0	81.6
0.6	0.8	-59.2 ± 0.0	16.8 ± 0.0	36.1 ± 0.0	31.0 ± 0.0	81.6
0.6	0.2	-58.8 ± 0.0	19.3 ± 0.0	36.5 ± 0.0	31.0 ± 0.0	81.6
0.4	0.6	-58.5 ± 1.3	17.9 ± 0.0	36.2 ± 0.0	30.1 ± 3.7	79.2
0.4	0.4	-58.3 ± 1.1	18.9 ± 0.0	36.5 ± 0.2	29.9 ± 3.9	78.7
0.2	0.8	-57.3 ± 2.5	15.8 ± 0.0	36.2 ± 0.8	28.8 ± 4.7	75.9
0.4	0.2	-55.3 ± 2.2	19.7 ± 0.0	35.4 ± 1.6	29.0 ± 4.2	76.4
0.2	0.6	-54.7 ± 3.3	16.9 ± 0.0	35.0 ± 1.6	27.9 ± 4.9	73.4
0.2	0.4	-51.6 ± 2.4	17.8 ± 0.0	33.6 ± 1.7	24.5 ± 6.1	64.6
0.2	0.2	-48.0 ± 2.7	17.6 ± 0.4	32.8 ± 2.1	21.8 ± 6.2	57.4

A.4.4 *mfold* results

Table A.14: *mfold* results for suboptimal foldings of *Haloarcula marismortui*

mfold ΔG **kcal/mol**	***efn2*** ΔG **kcal/mol**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
-59.5	-56.44	34	29	76.3

A.5 *Saccharomyces cerevisiae* 5S rRNA

A.5.1 SetPSO results, constant entropy

Table A.15: Experimental results for *Saccharomyces cerevisiae* 5S rRNA showing the mean and standard deviation over 30 samples.

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.2	0.4	0.6	-53.4 ± 0.0	8.7 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.2	0.4	0.8	-53.4 ± 0.0	8.4 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.2	0.6	0.4	-53.4 ± 0.0	8.0 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.2	0.6	0.6	-53.4 ± 0.0	7.6 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.2	0.6	0.8	-53.4 ± 0.0	7.3 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.2	0.8	0.4	-53.4 ± 0.0	7.0 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.2	0.8	0.6	-53.4 ± 0.0	6.7 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.2	0.8	0.8	-53.4 ± 0.0	6.4 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.2	0.4	-53.4 ± 0.0	10.4 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.2	0.6	-53.4 ± 0.0	9.5 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.2	0.8	-53.4 ± 0.0	8.7 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.4	0.4	-53.4 ± 0.0	9.2 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.4	0.6	-53.4 ± 0.0	8.4 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.4	0.8	-53.4 ± 0.0	7.8 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.6	0.4	-53.4 ± 0.0	8.4 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.6	0.6	-53.4 ± 0.0	7.9 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.6	0.8	-53.4 ± 0.0	7.1 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.8	0.2	-53.4 ± 0.0	9.8 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.8	0.4	-53.4 ± 0.0	7.7 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.8	0.6	-53.4 ± 0.0	7.2 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.4	0.8	0.8	-53.4 ± 0.0	6.3 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7

Continued on next page

Table A.15 – continued from previous page

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.6	0.2	0.2	-53.4 ± 0.0	10.8 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.2	0.4	-53.4 ± 0.0	9.1 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.2	0.6	-53.4 ± 0.0	7.7 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.2	0.8	-53.4 ± 0.0	7.2 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.4	0.2	-53.4 ± 0.0	10.4 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.4	0.4	-53.4 ± 0.0	8.2 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.4	0.6	-53.4 ± 0.0	7.7 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.4	0.8	-53.4 ± 0.0	6.7 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.6	0.2	-53.4 ± 0.0	9.7 ± 0.4	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.6	0.4	-53.4 ± 0.0	7.9 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.6	0.6	-53.4 ± 0.0	7.0 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.6	0.8	-53.4 ± 0.0	6.6 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.8	0.2	-53.4 ± 0.0	9.5 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.8	0.4	-53.4 ± 0.0	7.5 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.8	0.6	-53.4 ± 0.0	6.5 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.6	0.8	0.8	-53.4 ± 0.0	6.3 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.2	0.2	-53.4 ± 0.0	8.5 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.2	0.4	-53.4 ± 0.0	6.6 ± 0.2	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.2	0.6	-53.4 ± 0.0	5.6 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.2	0.8	-53.4 ± 0.0	5.5 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.4	0.2	-53.4 ± 0.0	8.1 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.4	0.4	-53.4 ± 0.0	6.5 ± 0.1	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.4	0.6	-53.4 ± 0.0	5.4 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.4	0.8	-53.4 ± 0.0	5.0 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.6	0.2	-53.4 ± 0.0	7.9 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.6	0.4	-53.4 ± 0.0	6.4 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.6	0.6	-53.4 ± 0.0	5.6 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7

Continued on next page

Table A.15 – continued from previous page

P_R	P_C	P_I	ΔG **kcal/mol**	**Swarm diversity**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
0.8	0.6	0.8	-53.4 ± 0.0	4.8 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.8	0.2	-53.4 ± 0.0	7.6 ± 0.6	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.8	0.4	-53.4 ± 0.0	6.3 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.8	0.6	-53.4 ± 0.0	5.2 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.8	0.8	0.8	-53.4 ± 0.0	5.0 ± 0.0	40.0 ± 0.0	28.0 ± 0.0	75.7
0.2	0.4	0.4	-53.4 ± 0.0	9.4 ± 0.0	40.0 ± 0.0	28.2 ± 0.0	76.1
0.2	0.8	0.2	-53.4 ± 0.0	8.9 ± 0.0	40.0 ± 0.0	28.2 ± 0.0	76.1
0.4	0.4	0.2	-53.4 ± 0.0	11.5 ± 0.0	39.9 ± 0.0	28.3 ± 0.8	76.6
0.4	0.2	0.2	-53.3 ± 0.0	12.2 ± 0.0	39.9 ± 0.0	28.7 ± 1.4	77.5
0.2	0.2	0.6	-53.3 ± 0.0	9.7 ± 0.0	39.8 ± 0.0	29.0 ± 1.7	78.4
0.2	0.2	0.8	-53.2 ± 0.0	8.8 ± 0.0	39.6 ± 0.0	29.8 ± 2.2	80.6
0.2	0.2	0.4	-53.1 ± 0.0	10.5 ± 0.0	39.5 ± 0.0	30.2 ± 2.3	81.5
0.4	0.6	0.2	-53.1 ± 0.8	10.4 ± 0.0	39.9 ± 0.0	27.9 ± 0.0	75.4
0.2	0.6	0.2	-52.7 ± 2.6	10.0 ± 0.0	39.5 ± 1.4	27.4 ± 3.6	74.1
0.2	0.4	0.2	-52.0 ± 2.3	11.2 ± 0.0	38.9 ± 1.5	28.0 ± 0.7	75.6
0.2	0.2	0.2	-50.0 ± 4.3	11.6 ± 0.0	38.2 ± 2.1	27.8 ± 3.9	75.1

A.5.2 SetPSO results, linear decreasing entropy

Table A.16: Experimental results for *Saccharomyces cerevisiae* 5S rRNA showing the mean and standard deviation over 30 samples.

		ΔG	**Swarm**	**Pairs**	**Pairs**	**%**
P_R	P_C	**kcal/mol**	**diversity**	**Predicted**	**Correct**	**Correct**
0.6	0.6	-54.1 ± 0.0	17.7 ± 0.0	42.0 ± 0.0	28.2 ± 0.0	76.1
0.4	0.8	-54.1 ± 0.0	17.5 ± 0.0	41.9 ± 0.0	28.3 ± 0.8	76.6
0.8	0.8	-54.0 ± 0.0	16.8 ± 0.0	41.9 ± 0.0	28.5 ± 1.1	77.0
0.8	0.2	-54.0 ± 0.0	16.7 ± 0.0	41.9 ± 0.0	28.5 ± 1.1	77.0
0.6	0.8	-54.0 ± 0.0	17.6 ± 0.0	41.9 ± 0.0	28.5 ± 1.1	77.0
0.2	0.8	-54.0 ± 0.0	16.4 ± 0.0	41.9 ± 0.0	28.5 ± 1.1	77.0
0.8	0.6	-54.0 ± 0.0	16.7 ± 0.0	41.9 ± 0.0	28.7 ± 1.4	77.5
0.8	0.4	-54.0 ± 0.0	16.6 ± 0.0	41.9 ± 0.0	28.7 ± 1.4	77.5
0.6	0.4	-54.0 ± 0.0	18.0 ± 0.0	41.9 ± 0.0	28.7 ± 1.4	77.5
0.4	0.6	-54.0 ± 0.0	17.8 ± 0.0	41.8 ± 0.0	29.0 ± 1.7	78.4
0.6	0.2	-54.0 ± 0.0	18.2 ± 0.0	41.8 ± 0.0	29.2 ± 1.9	78.8
0.4	0.4	-53.9 ± 0.0	18.2 ± 0.0	41.6 ± 0.0	30.0 ± 2.2	81.1
0.2	0.6	-53.7 ± 0.0	17.1 ± 0.0	41.3 ± 0.0	30.1 ± 2.2	81.3
0.4	0.2	-53.7 ± 0.0	18.2 ± 0.0	41.3 ± 0.0	30.6 ± 2.3	82.6
0.2	0.4	-52.9 ± 0.6	17.0 ± 0.0	40.2 ± 0.9	31.0 ± 2.3	83.9
0.2	0.2	-51.0 ± 1.8	16.8 ± 0.0	38.7 ± 1.5	29.9 ± 2.5	80.7

A.5.3 SetPSO results, minimum stem length of 2

Table A.17: Experimental results for *Saccharomyces cerevisiae* 5S rRNA with a minimum stem length of 2 showing the mean and standard deviation over 30 samples.

		ΔG	**Swarm**	**Pairs**	**Pairs**	**%**
P_R	P_C	**kcal/mol**	**diversity**	**Predicted**	**Correct**	**Correct**
0.6	0.6	-54.1 ± 0.0	17.7 ± 0.0	42.0 ± 0.0	28.2 ± 0.0	76.1
0.4	0.8	-54.1 ± 0.0	17.5 ± 0.0	41.9 ± 0.0	28.3 ± 0.8	76.6
0.8	0.8	-54.0 ± 0.0	16.8 ± 0.0	41.9 ± 0.0	28.5 ± 1.1	77.0
0.8	0.2	-54.0 ± 0.0	16.7 ± 0.0	41.9 ± 0.0	28.5 ± 1.1	77.0
0.6	0.8	-54.0 ± 0.0	17.6 ± 0.0	41.9 ± 0.0	28.5 ± 1.1	77.0
0.2	0.8	-54.0 ± 0.0	16.4 ± 0.0	41.9 ± 0.0	28.5 ± 1.1	77.0
0.8	0.6	-54.0 ± 0.0	16.7 ± 0.0	41.9 ± 0.0	28.7 ± 1.4	77.5
0.8	0.4	-54.0 ± 0.0	16.6 ± 0.0	41.9 ± 0.0	28.7 ± 1.4	77.5
0.6	0.4	-54.0 ± 0.0	18.0 ± 0.0	41.9 ± 0.0	28.7 ± 1.4	77.5
0.4	0.6	-54.0 ± 0.0	17.8 ± 0.0	41.8 ± 0.0	29.0 ± 1.7	78.4
0.6	0.2	-54.0 ± 0.0	18.2 ± 0.0	41.8 ± 0.0	29.2 ± 1.9	78.8
0.4	0.4	-53.9 ± 0.0	18.2 ± 0.0	41.6 ± 0.0	30.0 ± 2.2	81.1
0.2	0.6	-53.7 ± 0.0	17.1 ± 0.0	41.3 ± 0.0	30.1 ± 2.2	81.3
0.4	0.2	-53.7 ± 0.0	18.2 ± 0.0	41.3 ± 0.0	30.6 ± 2.3	82.6
0.2	0.4	-52.9 ± 0.6	17.0 ± 0.0	40.2 ± 0.9	31.0 ± 2.3	83.9
0.2	0.2	-51.0 ± 1.8	16.8 ± 0.0	38.7 ± 1.5	29.9 ± 2.5	80.7

A.5.4 *mfold* results

Table A.18: *mfold* results for suboptimal foldings of *Saccharomyces cerevisiae*

mfold ΔG **kcal/mol**	***efn2*** ΔG **kcal/mol**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
-53.5	-50.70	41	**33**	**89.2**
-53.0	-50.76	42	28	75.7

A.6 *Arthrobacter globiformis* 5S rRNA

A.6.1 SetPSO results, constant entropy

Table A.19: Experimental results for *Arthrobacter globiformis* 5S rRNA showing the mean and standard deviation over 30 samples.

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.4	0.4	0.8	-48.8 ± 0.0	8.9 ± 0.9	34.0 ± 0.0	17.0 ± 0.0	43.6
0.4	0.6	0.8	-48.8 ± 0.0	8.0 ± 0.7	34.0 ± 0.0	17.0 ± 0.0	43.6
0.4	0.8	0.6	-48.8 ± 0.0	7.6 ± 0.7	34.0 ± 0.0	17.0 ± 0.0	43.6
0.6	0.4	0.8	-48.8 ± 0.0	8.2 ± 0.7	34.0 ± 0.0	17.0 ± 0.0	43.6
0.6	0.6	0.6	-48.8 ± 0.0	8.1 ± 1.0	34.0 ± 0.0	17.0 ± 0.0	43.6
0.6	0.6	0.8	-48.8 ± 0.0	7.3 ± 0.8	34.0 ± 0.0	17.0 ± 0.0	43.6
0.6	0.8	0.6	-48.8 ± 0.0	7.3 ± 0.7	34.0 ± 0.0	17.0 ± 0.0	43.6
0.6	0.8	0.8	-48.8 ± 0.0	6.8 ± 0.7	34.0 ± 0.0	17.0 ± 0.0	43.6
0.2	0.8	0.6	-48.8 ± 0.1	7.5 ± 0.8	34.0 ± 0.0	17.1 ± 0.4	43.8
0.4	0.6	0.6	-48.8 ± 0.1	8.4 ± 0.9	34.0 ± 0.0	17.1 ± 0.4	43.8
0.4	0.8	0.8	-48.8 ± 0.1	7.4 ± 0.8	34.0 ± 0.0	17.1 ± 0.4	43.8
0.6	0.2	0.8	-48.8 ± 0.1	8.9 ± 0.7	34.0 ± 0.0	17.1 ± 0.4	43.8
0.6	0.4	0.6	-48.8 ± 0.1	8.7 ± 0.6	34.0 ± 0.0	17.1 ± 0.4	43.8
0.8	0.8	0.8	-48.8 ± 0.1	6.2 ± 0.9	34.0 ± 0.0	17.1 ± 0.4	43.8
0.2	0.6	0.8	-48.8 ± 0.1	7.9 ± 0.6	34.0 ± 0.0	17.1 ± 0.5	43.9
0.6	0.6	0.4	-48.8 ± 0.1	9.0 ± 0.9	34.0 ± 0.0	17.1 ± 0.5	43.9
0.2	0.8	0.8	-48.8 ± 0.2	7.3 ± 0.8	34.1 ± 0.6	17.0 ± 0.0	43.6
0.4	0.2	0.8	-48.8 ± 0.1	9.7 ± 0.7	34.0 ± 0.0	17.2 ± 0.6	44.1
0.4	0.4	0.6	-48.8 ± 0.1	9.4 ± 0.6	34.0 ± 0.0	17.2 ± 0.6	44.1
0.4	0.8	0.4	-48.8 ± 0.2	8.5 ± 0.9	34.1 ± 0.7	17.0 ± 0.0	43.6
0.6	0.8	0.4	-48.8 ± 0.1	8.0 ± 1.0	34.0 ± 0.0	17.2 ± 0.6	44.1

Continued on next page

Table A.19 – continued from previous page

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.8	0.4	0.6	-48.8 ± 0.1	7.5 ± 0.7	34.0 ± 0.0	17.2 ± 0.6	44.1
0.8	0.6	0.6	-48.8 ± 0.1	6.7 ± 0.6	34.0 ± 0.0	17.3 ± 0.7	44.3
0.8	0.8	0.6	-48.8 ± 0.1	6.6 ± 0.9	34.0 ± 0.0	17.2 ± 0.6	44.1
0.2	0.6	0.4	-48.8 ± 0.1	8.9 ± 0.8	34.0 ± 0.0	17.2 ± 0.6	44.1
0.6	0.2	0.6	-48.8 ± 0.1	9.1 ± 0.7	34.0 ± 0.0	17.3 ± 0.8	44.4
0.8	0.2	0.8	-48.8 ± 0.1	7.7 ± 0.7	34.0 ± 0.0	17.3 ± 0.8	44.4
0.8	0.6	0.8	-48.8 ± 0.1	6.6 ± 0.6	34.0 ± 0.0	17.3 ± 0.8	44.4
0.2	0.4	0.6	-48.8 ± 0.1	9.5 ± 0.5	34.0 ± 0.0	17.2 ± 0.6	44.1
0.2	0.4	0.8	-48.8 ± 0.1	9.0 ± 0.6	34.0 ± 0.0	17.4 ± 0.8	44.6
0.2	0.6	0.6	-48.8 ± 0.2	8.3 ± 0.7	34.1 ± 0.7	17.1 ± 0.5	43.9
0.4	0.4	0.4	-48.8 ± 0.1	9.9 ± 0.6	34.0 ± 0.0	17.3 ± 0.8	44.4
0.8	0.4	0.8	-48.8 ± 0.1	6.9 ± 0.7	34.0 ± 0.0	17.4 ± 0.8	44.6
0.4	0.6	0.4	-48.7 ± 0.2	9.1 ± 0.8	34.1 ± 0.6	17.3 ± 0.7	44.3
0.6	0.2	0.4	-48.7 ± 0.1	10.0 ± 0.6	34.0 ± 0.0	17.5 ± 0.9	44.8
0.4	0.2	0.6	-48.7 ± 0.2	9.8 ± 0.6	34.1 ± 0.7	17.3 ± 0.8	44.4
0.8	0.4	0.4	-48.7 ± 0.1	8.2 ± 0.8	34.0 ± 0.0	17.5 ± 0.9	44.9
0.6	0.4	0.4	-48.7 ± 0.2	9.3 ± 0.6	34.1 ± 0.7	17.4 ± 0.8	44.6
0.2	0.8	0.4	-48.7 ± 0.2	7.7 ± 1.0	34.2 ± 0.9	17.2 ± 0.6	44.1
0.4	0.2	0.4	-48.7 ± 0.1	10.6 ± 0.6	34.0 ± 0.0	17.6 ± 0.9	45.1
0.2	0.4	0.4	-48.7 ± 0.2	10.1 ± 0.5	34.1 ± 0.7	17.5 ± 0.9	44.9
0.8	0.8	0.4	-48.7 ± 0.1	7.4 ± 0.8	34.0 ± 0.0	17.7 ± 1.0	45.5
0.8	0.6	0.4	-48.7 ± 0.2	7.6 ± 0.9	34.1 ± 0.7	17.6 ± 0.9	45.1
0.8	0.2	0.6	-48.7 ± 0.1	8.1 ± 0.6	34.0 ± 0.0	18.0 ± 1.0	46.2
0.8	0.4	0.2	-48.7 ± 0.2	9.5 ± 1.1	34.1 ± 0.7	17.8 ± 1.0	45.6
0.8	0.2	0.4	-48.6 ± 0.1	8.4 ± 0.6	34.0 ± 0.0	18.2 ± 1.0	46.7
0.6	0.2	0.2	-48.6 ± 0.2	11.0 ± 0.7	34.3 ± 1.0	17.7 ± 1.0	45.3
0.6	0.8	0.2	-48.6 ± 0.3	9.7 ± 1.0	34.5 ± 1.3	17.5 ± 0.9	44.9

Continued on next page

Table A.19 – continued from previous page

P_R	P_C	P_I	ΔG **kcal/mol**	**Swarm diversity**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
0.4	0.8	0.2	-48.6 ± 0.4	9.9 ± 1.0	34.7 ± 1.5	17.3 ± 0.7	44.3
0.8	0.2	0.2	-48.6 ± 0.2	9.4 ± 0.6	34.1 ± 0.7	18.3 ± 1.0	46.8
0.8	0.6	0.2	-48.6 ± 0.2	9.2 ± 1.2	34.3 ± 1.0	18.1 ± 1.0	46.3
0.6	0.4	0.2	-48.6 ± 0.3	10.6 ± 0.8	34.4 ± 1.2	17.8 ± 1.0	45.6
0.8	0.8	0.2	-48.6 ± 0.3	9.2 ± 1.3	34.5 ± 1.4	17.9 ± 1.0	45.8
0.4	0.6	0.2	-48.5 ± 0.4	10.6 ± 0.7	35.1 ± 1.8	17.3 ± 0.8	44.4
0.6	0.6	0.2	-48.5 ± 0.4	10.2 ± 1.0	35.2 ± 1.8	17.5 ± 0.9	44.8
0.4	0.4	0.2	-48.4 ± 0.3	10.8 ± 0.7	34.8 ± 1.6	17.7 ± 1.0	45.3
0.2	0.8	0.2	-48.4 ± 0.4	9.1 ± 1.0	35.3 ± 1.7	17.1 ± 0.5	43.9
0.2	0.6	0.2	-48.4 ± 0.4	10.1 ± 0.5	35.5 ± 1.9	17.5 ± 0.9	44.8
0.4	0.2	0.2	-48.3 ± 0.4	11.7 ± 0.6	35.2 ± 2.1	17.7 ± 2.1	45.4
0.2	0.4	0.2	-48.3 ± 0.5	10.8 ± 0.7	35.5 ± 1.9	17.1 ± 0.5	43.9
0.2	0.2	0.6	-48.1 ± 0.5	10.2 ± 0.6	34.1 ± 2.3	17.3 ± 2.3	44.4
0.2	0.2	0.8	-48.1 ± 0.6	9.8 ± 0.8	33.1 ± 2.1	18.1 ± 3.7	46.4
0.2	0.2	0.4	-47.9 ± 0.6	10.6 ± 0.5	34.1 ± 2.7	17.4 ± 3.2	44.6
0.2	0.2	0.2	-47.4 ± 1.2	11.6 ± 0.7	35.2 ± 2.5	17.4 ± 2.3	44.7

A.7 *Caenorhabditis elegans* 16S rRNA

A.7.1 SetPSO results, constant entropy

Table A.20: Experimental results for *Caenorhabditis elegans* 16S rRNA showing the mean and standard deviation over 30 samples.

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.8	0.4	0.8	-100.7 ± 3.0	56.1 ± 2.5	218.4 ± 7.4	28.3 ± 4.3	15.0
0.8	0.6	0.8	-100.7 ± 4.1	57.0 ± 3.6	216.7 ± 4.1	24.8 ± 5.1	13.1
0.8	0.2	0.8	-100.6 ± 2.9	56.8 ± 3.1	216.5 ± 4.8	26.6 ± 6.1	14.1
0.8	0.8	0.8	-99.4 ± 4.2	55.8 ± 4.2	216.3 ± 4.4	27.1 ± 7.2	14.3
0.8	0.2	0.6	-94.9 ± 3.6	62.3 ± 2.8	217.4 ± 7.1	27.3 ± 5.9	14.4
0.8	0.6	0.6	-92.7 ± 5.2	65.2 ± 2.7	217.2 ± 6.0	23.5 ± 6.5	12.4
0.8	0.4	0.6	-92.5 ± 4.9	64.1 ± 3.3	216.0 ± 5.3	29.4 ± 6.9	15.6
0.8	0.8	0.6	-90.8 ± 5.5	65.3 ± 3.9	216.3 ± 5.3	25.8 ± 6.6	13.7
0.6	0.2	0.8	-87.4 ± 4.3	63.1 ± 1.7	207.0 ± 4.9	25.0 ± 7.7	13.2
0.6	0.6	0.8	-87.0 ± 5.3	63.6 ± 2.4	209.2 ± 6.2	22.3 ± 7.3	11.8
0.6	0.4	0.8	-86.4 ± 4.3	64.2 ± 2.5	207.9 ± 5.6	24.7 ± 5.6	13.1
0.6	0.8	0.8	-84.7 ± 5.1	63.9 ± 2.3	209.1 ± 7.1	27.4 ± 8.1	14.5
0.8	0.2	0.4	-83.2 ± 4.3	70.4 ± 2.3	216.9 ± 6.0	25.0 ± 5.1	13.2
0.8	0.4	0.4	-79.2 ± 5.5	71.9 ± 3.0	216.8 ± 5.4	24.0 ± 6.5	12.7
0.6	0.2	0.6	-79.1 ± 4.0	67.5 ± 3.1	204.9 ± 7.4	26.3 ± 7.8	13.9
0.8	0.6	0.4	-76.5 ± 5.6	74.0 ± 2.3	215.7 ± 7.1	25.1 ± 8.1	13.3
0.6	0.4	0.6	-73.9 ± 5.8	69.5 ± 2.3	205.8 ± 7.1	21.4 ± 8.4	11.3
0.8	0.8	0.4	-73.8 ± 6.3	74.1 ± 2.9	215.2 ± 6.3	25.5 ± 7.8	13.5
0.6	0.6	0.6	-73.0 ± 4.6	70.2 ± 1.9	206.6 ± 6.7	22.8 ± 8.1	12.1
0.6	0.8	0.6	-72.1 ± 6.4	70.9 ± 2.6	207.0 ± 6.4	22.5 ± 7.5	11.9
0.4	0.2	0.8	-71.4 ± 4.2	64.5 ± 2.0	192.9 ± 6.7	20.9 ± 6.9	11.1

Continued on next page

Table A.20 – continued from previous page

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.4	0.6	0.8	-69.6 ± 5.7	66.0 ± 2.0	195.6 ± 7.5	19.7 ± 6.0	10.4
0.4	0.8	0.8	-69.5 ± 4.4	65.7 ± 1.9	193.7 ± 7.4	23.5 ± 8.1	12.4
0.4	0.4	0.8	-68.7 ± 4.4	65.6 ± 2.2	194.4 ± 5.6	23.8 ± 6.2	12.6
0.6	0.2	0.4	-61.3 ± 5.2	76.4 ± 2.4	207.6 ± 7.9	21.0 ± 8.5	11.1
0.4	0.2	0.6	-61.1 ± 5.0	69.5 ± 2.2	191.1 ± 8.2	20.4 ± 7.5	10.8
0.8	0.2	0.2	-59.8 ± 6.9	81.7 ± 2.8	215.0 ± 7.9	21.1 ± 9.6	11.2
0.4	0.4	0.6	-57.4 ± 4.2	70.8 ± 2.4	194.3 ± 6.5	21.6 ± 6.6	11.4
0.6	0.4	0.4	-56.6 ± 4.9	76.5 ± 2.3	204.6 ± 6.1	22.4 ± 7.0	11.9
0.6	0.6	0.4	-56.0 ± 5.5	77.9 ± 2.1	205.5 ± 6.5	19.3 ± 5.6	10.2
0.8	0.4	0.2	-54.8 ± 6.3	82.2 ± 2.4	213.1 ± 7.0	18.7 ± 7.5	9.9
0.4	0.6	0.6	-54.0 ± 4.2	71.2 ± 2.2	192.8 ± 8.6	20.3 ± 8.4	10.7
0.4	0.8	0.6	-53.5 ± 5.4	71.6 ± 2.4	195.5 ± 8.8	20.8 ± 7.5	11.0
0.6	0.8	0.4	-52.3 ± 6.4	78.1 ± 1.7	204.0 ± 7.8	19.9 ± 6.9	10.5
0.2	0.2	0.8	-51.1 ± 4.0	61.6 ± 1.9	173.5 ± 8.5	15.9 ± 7.4	8.4
0.2	0.4	0.8	-50.1 ± 4.7	62.1 ± 2.1	171.7 ± 6.1	18.9 ± 8.1	10.0
0.8	0.6	0.2	-50.0 ± 7.8	83.7 ± 2.1	213.5 ± 7.0	16.9 ± 6.5	8.9
0.2	0.6	0.8	-49.6 ± 5.6	62.7 ± 1.9	174.3 ± 8.7	17.1 ± 7.5	9.1
0.2	0.8	0.8	-48.9 ± 4.7	62.1 ± 2.1	172.7 ± 11.2	16.3 ± 7.3	8.6
0.8	0.8	0.2	-47.0 ± 6.7	83.6 ± 1.9	214.8 ± 7.1	22.3 ± 7.1	11.8
0.4	0.2	0.4	-44.2 ± 6.9	75.5 ± 2.3	192.9 ± 7.1	18.8 ± 9.3	10.0
0.2	0.2	0.6	-40.1 ± 4.9	65.8 ± 2.4	171.1 ± 6.4	17.1 ± 7.9	9.0
0.4	0.4	0.4	-39.3 ± 5.4	76.6 ± 2.5	191.5 ± 6.8	18.4 ± 8.6	9.7
0.2	0.4	0.6	-37.8 ± 4.4	65.7 ± 1.9	170.3 ± 8.8	18.7 ± 7.0	9.9
0.2	0.6	0.6	-37.2 ± 5.6	67.2 ± 1.7	173.7 ± 7.2	17.3 ± 6.7	9.2
0.4	0.6	0.4	-35.6 ± 5.1	77.2 ± 2.0	190.3 ± 7.9	18.0 ± 8.1	9.5
0.2	0.8	0.6	-34.9 ± 5.1	66.6 ± 2.5	171.8 ± 10.1	14.7 ± 7.2	7.8
0.6	0.2	0.2	-34.6 ± 5.3	84.9 ± 2.1	202.0 ± 6.0	16.7 ± 8.6	8.8

Continued on next page

Table A.20 – continued from previous page

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.4	0.8	0.4	-34.5 ± 6.5	77.5 ± 2.7	190.7 ± 8.2	18.4 ± 7.2	9.8
0.6	0.4	0.2	-31.9 ± 7.4	85.9 ± 2.3	203.0 ± 6.9	15.4 ± 10.0	8.2
0.2	0.2	0.4	-26.1 ± 6.3	70.5 ± 2.0	170.2 ± 6.2	15.9 ± 7.9	8.4
0.6	0.6	0.2	-25.9 ± 5.4	85.8 ± 2.1	203.1 ± 7.4	17.2 ± 8.6	9.1
0.6	0.8	0.2	-21.7 ± 6.0	86.7 ± 2.1	204.8 ± 7.6	16.0 ± 7.9	8.5
0.2	0.4	0.4	-21.1 ± 5.3	70.9 ± 2.5	166.9 ± 10.5	17.3 ± 6.6	9.1
0.2	0.8	0.4	-20.4 ± 5.7	72.1 ± 2.3	172.1 ± 8.5	14.9 ± 7.8	7.9
0.2	0.6	0.4	-19.7 ± 4.6	72.2 ± 2.2	170.2 ± 8.1	14.9 ± 7.2	7.9
0.4	0.2	0.2	-16.5 ± 5.5	84.0 ± 2.1	190.4 ± 7.3	13.4 ± 5.6	7.1
0.4	0.4	0.2	-14.7 ± 5.3	84.7 ± 2.1	190.8 ± 6.9	17.1 ± 8.1	9.0
0.4	0.6	0.2	-8.1 ± 5.3	85.3 ± 2.4	190.3 ± 7.1	15.8 ± 8.5	8.3
0.4	0.8	0.2	-7.2 ± 5.7	85.4 ± 2.7	190.1 ± 7.9	14.7 ± 7.5	7.8
0.2	0.2	0.2	-0.5 ± 5.1	78.9 ± 1.8	170.9 ± 7.0	11.7 ± 6.0	6.2
0.2	0.6	0.2	0.7 ± 6.2	80.2 ± 2.5	171.4 ± 9.7	12.9 ± 7.8	6.8
0.2	0.8	0.2	1.7 ± 5.8	80.3 ± 2.0	174.7 ± 7.5	14.6 ± 6.7	7.7
0.2	0.4	0.2	2.5 ± 5.9	79.9 ± 2.1	172.2 ± 6.9	11.6 ± 8.3	6.1

Table A.21: *mfold* results for suboptimal foldings of *Caenorhabditis elegans* 16S rRNA

mfold ΔG **kcal/mol**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
-128.97	216	20	10.6
-126.56	221	25	13.2
-126.18	221	25	13.2
-125.99	213	20	10.6
-125.22	217	40	21.2
-124.20	211	40	21.2
-124.04	216	40	21.2
-123.68	212	37	19.6
-123.20	219	32	16.9
-122.56	218	35	18.5
-122.10	211	20	10.6
-121.59	213	40	21.2
-121.46	219	25	13.2
-120.81	216	37	19.6
-120.06	208	35	18.5
-118.90	211	27	14.3
-118.87	205	27	14.3
-117.46	206	32	16.9
-115.94	200	27	14.3

A.8 *Homo sapiens* 16S rRNA

A.8.1 SetPSO results, constant entropy

Table A.22: Experimental results for *Homo sapiens* 16S rRNA showing the mean and standard deviation over 30 samples.

			ΔG	Swarm	Pairs	Pairs	%
P_R	P_C	P_I	kcal/mol	diversity	Predicted	Correct	Correct
0.8	0.6	0.8	-191.1 ± 6.4	72.0 ± 3.3	234.5 ± 6.2	54.9 ± 15.1	20.6
0.8	0.4	0.8	-190.4 ± 5.1	72.7 ± 3.5	236.8 ± 6.5	55.8 ± 12.3	20.9
0.8	0.8	0.8	-190.2 ± 5.5	70.3 ± 4.3	237.8 ± 7.0	53.1 ± 12.5	19.9
0.8	0.2	0.8	-188.2 ± 3.7	72.7 ± 3.4	233.2 ± 7.2	52.6 ± 11.9	19.7
0.8	0.4	0.6	-183.3 ± 5.8	79.5 ± 3.6	237.7 ± 7.6	51.1 ± 13.9	19.1
0.8	0.8	0.6	-181.9 ± 6.4	79.7 ± 2.9	237.6 ± 5.0	55.2 ± 15.2	20.7
0.8	0.2	0.6	-180.6 ± 3.2	78.5 ± 3.5	236.1 ± 5.5	54.3 ± 10.7	20.3
0.8	0.6	0.6	-179.9 ± 6.1	80.2 ± 2.8	239.8 ± 7.0	57.8 ± 12.3	21.7
0.6	0.2	0.8	-174.2 ± 3.7	78.6 ± 1.9	228.3 ± 7.2	50.1 ± 11.7	18.8
0.6	0.4	0.8	-173.1 ± 5.6	78.9 ± 3.2	228.3 ± 6.9	47.5 ± 13.6	17.8
0.6	0.8	0.8	-172.4 ± 5.3	79.0 ± 3.4	230.0 ± 7.6	45.0 ± 9.4	16.9
0.6	0.6	0.8	-171.7 ± 6.6	79.9 ± 2.5	228.3 ± 6.1	47.2 ± 10.4	17.7
0.8	0.2	0.4	-168.9 ± 4.8	85.9 ± 2.9	239.6 ± 7.0	55.2 ± 13.9	20.7
0.8	0.4	0.4	-168.2 ± 7.9	86.7 ± 2.9	240.7 ± 6.7	48.7 ± 11.7	18.2
0.8	0.6	0.4	-167.8 ± 8.7	87.8 ± 2.5	239.4 ± 7.1	48.7 ± 12.4	18.2
0.8	0.8	0.4	-167.0 ± 7.5	88.3 ± 3.1	242.2 ± 5.1	51.8 ± 12.9	19.4
0.6	0.2	0.6	-162.7 ± 4.0	83.7 ± 1.9	227.7 ± 6.5	51.4 ± 12.8	19.2
0.6	0.4	0.6	-159.1 ± 5.5	85.0 ± 2.6	227.4 ± 6.6	47.5 ± 13.0	17.8
0.6	0.8	0.6	-157.1 ± 5.3	85.9 ± 2.4	230.1 ± 7.8	46.2 ± 10.8	17.3
0.6	0.6	0.6	-156.4 ± 7.4	85.9 ± 2.5	227.3 ± 7.5	46.2 ± 12.3	17.3
0.4	0.2	0.8	-153.6 ± 4.3	80.0 ± 2.9	213.8 ± 6.4	42.1 ± 13.4	15.8

Continued on next page

Table A.22 – continued from previous page

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.4	0.4	0.8	-149.5 ± 5.5	79.9 ± 1.5	212.0 ± 6.1	40.6 ± 11.5	15.2
0.4	0.8	0.8	-149.5 ± 6.0	79.2 ± 2.2	215.5 ± 8.0	41.5 ± 8.5	15.6
0.8	0.2	0.2	-147.9 ± 5.1	95.2 ± 2.7	240.8 ± 7.4	42.4 ± 13.0	15.9
0.4	0.6	0.8	-147.7 ± 6.3	80.3 ± 2.9	214.1 ± 7.6	39.8 ± 11.8	14.9
0.6	0.2	0.4	-147.0 ± 4.8	90.7 ± 2.4	230.3 ± 6.2	47.3 ± 10.6	17.7
0.8	0.4	0.2	-143.0 ± 6.2	96.0 ± 3.2	244.4 ± 7.9	40.7 ± 11.5	15.2
0.8	0.6	0.2	-142.9 ± 6.3	94.8 ± 3.2	246.2 ± 6.4	41.9 ± 12.9	15.7
0.6	0.4	0.4	-142.2 ± 5.5	91.4 ± 2.5	226.2 ± 7.1	44.7 ± 11.9	16.8
0.6	0.6	0.4	-139.7 ± 8.4	92.0 ± 2.7	232.8 ± 7.5	40.5 ± 11.0	15.2
0.4	0.2	0.6	-138.9 ± 4.4	84.3 ± 2.7	210.6 ± 7.2	36.7 ± 13.1	13.7
0.6	0.8	0.4	-138.6 ± 6.9	91.7 ± 2.4	228.6 ± 7.7	42.2 ± 12.8	15.8
0.8	0.8	0.2	-136.4 ± 4.7	96.1 ± 2.9	245.6 ± 7.6	37.8 ± 11.4	14.2
0.4	0.6	0.6	-134.3 ± 4.3	85.5 ± 1.9	212.8 ± 6.9	39.7 ± 9.8	14.9
0.4	0.4	0.6	-133.8 ± 6.1	85.6 ± 3.0	212.8 ± 8.0	40.2 ± 13.0	15.1
0.4	0.8	0.6	-133.0 ± 4.3	85.8 ± 2.4	215.2 ± 8.7	40.3 ± 9.6	15.1
0.6	0.2	0.2	-122.8 ± 5.3	100.0 ± 2.3	233.1 ± 7.1	35.7 ± 12.4	13.4
0.4	0.2	0.4	-122.7 ± 3.9	90.6 ± 2.4	213.0 ± 7.4	36.8 ± 12.9	13.8
0.2	0.2	0.8	-121.5 ± 6.4	73.9 ± 2.3	186.2 ± 8.3	31.1 ± 12.2	11.6
0.2	0.8	0.8	-119.9 ± 7.6	75.0 ± 2.8	192.8 ± 8.2	37.5 ± 10.1	14.0
0.6	0.4	0.2	-119.9 ± 5.9	100.0 ± 3.0	233.9 ± 6.7	35.7 ± 10.8	13.4
0.2	0.4	0.8	-118.4 ± 6.1	76.3 ± 2.7	189.9 ± 7.1	30.5 ± 10.3	11.4
0.4	0.4	0.4	-118.0 ± 4.5	91.4 ± 3.1	214.5 ± 8.7	38.1 ± 10.8	14.3
0.6	0.8	0.2	-117.6 ± 6.1	98.8 ± 2.4	235.1 ± 6.6	36.3 ± 14.6	13.6
0.6	0.6	0.2	-117.5 ± 6.5	99.4 ± 2.8	236.3 ± 5.3	35.7 ± 7.7	13.4
0.4	0.6	0.4	-117.3 ± 6.5	90.8 ± 2.4	214.1 ± 8.9	38.8 ± 10.0	14.5
0.2	0.6	0.8	-117.1 ± 5.9	76.6 ± 2.7	190.7 ± 8.3	32.2 ± 11.4	12.1
0.4	0.8	0.4	-115.3 ± 5.5	91.7 ± 2.2	217.1 ± 6.4	33.8 ± 10.7	12.7

Continued on next page

Table A.22 – continued from previous page

P_R	P_C	P_I	ΔG kcal/mol	Swarm diversity	Pairs Predicted	Pairs Correct	% Correct
0.2	0.2	0.6	-108.9 ± 5.3	79.1 ± 2.7	189.6 ± 7.3	29.9 ± 8.8	11.2
0.2	0.8	0.6	-108.0 ± 7.2	79.5 ± 2.4	192.3 ± 7.4	32.2 ± 12.5	12.1
0.2	0.4	0.6	-106.7 ± 5.6	79.4 ± 2.7	187.1 ± 8.2	33.1 ± 13.1	12.4
0.2	0.6	0.6	-105.3 ± 5.2	79.8 ± 2.4	190.3 ± 7.2	33.8 ± 11.8	12.7
0.4	0.8	0.2	-100.1 ± 7.3	98.3 ± 2.9	228.8 ± 8.4	31.2 ± 7.8	11.7
0.4	0.2	0.2	-98.9 ± 4.5	98.8 ± 3.0	218.2 ± 7.1	27.7 ± 11.6	10.4
0.4	0.4	0.2	-97.2 ± 5.2	99.9 ± 2.3	219.6 ± 7.4	28.3 ± 9.5	10.6
0.4	0.6	0.2	-96.4 ± 5.8	99.8 ± 1.9	221.4 ± 7.9	30.3 ± 11.1	11.3
0.2	0.8	0.4	-95.2 ± 8.1	84.4 ± 2.3	195.5 ± 8.2	29.5 ± 9.2	11.0
0.2	0.2	0.4	-92.0 ± 5.9	83.9 ± 2.3	188.1 ± 9.2	28.9 ± 10.7	10.8
0.2	0.4	0.4	-91.9 ± 5.8	85.5 ± 2.9	191.9 ± 9.7	30.2 ± 9.9	11.3
0.2	0.6	0.4	-91.6 ± 7.1	85.8 ± 3.1	193.7 ± 9.9	30.3 ± 9.0	11.3
0.2	0.8	0.2	-81.4 ± 5.8	92.3 ± 3.1	208.7 ± 10.9	26.2 ± 6.8	9.8
0.2	0.6	0.2	-78.0 ± 6.0	92.9 ± 2.4	200.9 ± 6.9	24.8 ± 8.9	9.3
0.2	0.4	0.2	-77.0 ± 5.3	93.7 ± 2.7	199.4 ± 9.7	22.9 ± 9.1	8.6
0.2	0.2	0.2	-75.5 ± 5.4	93.4 ± 2.3	193.3 ± 7.2	23.2 ± 8.3	8.7

Table A.23: *mfold* results for suboptimal foldings of *Homo sapiens* 16S rRNA

mfold ΔG **kcal/mol**	**Pairs Predicted**	**Pairs Correct**	**% Correct**
-222.51	255	93	34.8
-219.97	251	53	19.9
-219.87	259	52	19.5
-217.55	255	59	22.1
-217.20	258	97	36.3
-214.50	256	84	31.5
-213.14	262	44	16.5
-211.12	258	67	25.1
-210.39	255	84	31.5
-210.26	257	61	22.8
-207.98	260	52	19.5
-207.85	257	83	31.1
-206.53	256	37	13.9
-205.51	262	43	16.1
-204.14	266	57	21.3
-202.39	245	53	19.9
-198.70	259	50	18.7
-198.17	258	44	16.5

Appendix B

Acronyms

CE	Constant Entropy
CILib	Computational Intelligence Library
DNA	Deoxyribonucleic acid
DPA	Dynamic Programming Algorithm
EC	Evolutionary Computation
GA	Genetic Algorithm
INN	Individual Nearest Neighbour
INN-HB	Individual Nearest Neighbour Hydrogen Bonding model
LDE	Linear Decreasing Entropy
mRNA	Messenger Ribonucleic acid
NMR	Nuclear magnetic resonance
PSO	Particle Swarm Optimiser
RNA	Ribonucleic acid
rRNA	Ribosomal Ribonucleic acid
SetPSO	Set Particle Swarm Optimiser
tRNA	Transfer Ribonucleic acid
VM	Virtual Machine

Appendix C

Symbols

This appendix lists the mathematical symbols used throughout this dissertation, and their definitions. The symbols used within each chapter are listed under separate sections. Each section lists only newly introduced symbols, and those that have by necessity been re-defined. In cases where an equation in the text defines a symbol, the relevant equation and page numbers are provided in brackets, after the symbol definition:

C.1 Chapter 2: Ribonucleic acid

$[i, j]$	Base pairing formed between base at position i and base at position j
m	Minimum length of hairpin loop [Eq. (2.1), pg. 10]
S	Set containing all possible stems
C	Subset of S defining a conformation
ΔG	Change in free energy of an RNA conformation

C.2 Chapter 3: RNA thermodynamics

ΔG	Change in free energy of an RNA conformation [Eq. (3.1), pg. 18]
ΔH	Change in enthalpy, the amount of energy posessed by a thermodynamic system to transfer between itself and the environment
T	Temperature

ΔS	Change in entropy, a measure of randomness or disorder of a system
Q	Partition function [Eq. (3.2), pg. 19]
S	Structure or folding
ΔG_S	Change in free energy of structure S
R	Gas constant
r_i	Nucleotide at position i
r_j	Nucleotide at position j

C.3 Chapter 4: Particle Swarm Optimisation

$\mathbf{x}$	Position vector of particle
$\mathbf{v}$	Velocity vector of particle
$\mathbf{x}_i(t)$	i-th dimension of the particle position vector $\mathbf{x}$ at time step t [Eq. (4.1), pg. 25]
$v_{ij}(t)$	The velocity of particle i in dimension $j = 1, ..., n_x$ at time step t
$x_{ij}(t)$	The position of particle i in dimension $j = 1, ..., n_x$ at time step t
c_1	Positive acceleration constant
c_2	Positive acceleration constant
$r_{1j}(t)$	Random number sampled from $\sim U(0, 1)$
$r_{2j}(t)$	Random number sampled from $\sim U(0, 1)$
$\mathbf{y}_i$	Particle i's personal best position ($pbest$)
$\hat{\mathbf{y}}(t)$	Global best position at time step t [Eq. (4.4), pg. 26]
n_s	Number of particles in swarm
ω	Inertia weight

C.4 Chapter 5: Set Particle Swarm Optimisation

U	Universal set containing all possible elements
O	open set, elements to be removed from the current particle position

C	close set, stems to be added to the new particle position
B	temporary set
e	element from U
$S.X_i$	position X of particle i in the swarm S
$S.Y_i$	personal best position Y of particle i in the swarm S
$S.\widehat{Y}$	neighbourhood best position of swarm S
P_I	Entropy weight
P_C	Closing probability
P_R	Random add probability
$\mathbf{D}$	Diversity [Eq. (5.4), pg. 42]

Appendix D

Derived Publications

This appendix lists the publications derived from the work presented here. Both accepted publications and those in the process of submission at the time of this dissertation's publication are listed:

- C. Marais Neethling and Andries P. Engelbrecht. Determining RNA Secondary Structure using Set-based Particle Swarm Optimization. In Gary G. Yen, Simon M. Lucas, Gary Fogel, Graham Kendall, Ralf Salomon, Byoung-Tak Zhang, Carlos A. Coello Coello, Thomas Philip Runarsson, editors, *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 1670–1677, Vancouver, BC, Canada, 16-21 July 2006. IEEE Press