# Computational Design and Experimental Validation of Functional Ribonucleic Acid Nanostructures

Kasra Zandi

A Thesis
In The Department
Of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Computer Science) at
Concordia University

January 2018

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr. Kasra Zandi**

Entitled: **Computational Design and Experimental Validation of Functional Ribonucleic Acid Nanostructures**

and submitted in partial fulfillment of the requirements for the degree of

**Doctor of Philosophy (Computer Science)**

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Lyes Kadem Chair

Dr. Francois Major External Examiner

Dr. Paul Joyce Examiner

Dr. Adam Krzyzak Examiner

Dr. Sudhir Mudur Examiner

Dr. Nawwaf Kharma Supervisor

Dr. Gregory Butler Supervisor

Approved by: ______________________________

Dr. Volker Harslev Chair of Department or Graduate Program Director

Feb, 26th 2018 ______________________________

Dr. Amir Asif Dean

Faculty of Engineering and Computer Science

# Abstract

Computational Design and Experimental Validation of Functional Ribonucleic Acid Nanostructures

**Kasra Zandi, Ph.D.**
**Concordia University, 2018**

In living cells, two major classes of ribonucleic acid (RNA) molecules can be found. The first class called the messenger RNA (mRNA) contains the genetic information that allows the ribosome to read and translate it into proteins. The second class called non-coding RNA (ncRNA), do not code for proteins and are involved with key cellular processes, such as gene expression regulation, splicing, differentiation and development. NcRNAs fold into an ensemble of thermodynamically stable secondary structures, which will eventually lead the molecule to fold into a specific 3D structure. It is widely known that ncRNAs carry their functions via their 3D structures as well as their molecular composition. The secondary structure of ncRNAs is composed of different types of structural elements (motifs) such as stacking base pairs, internal loops, hairpin loops and pseudoknots. Pseudoknots are specifically difficult to model, are abundant in nature and known to stabilize the functional form of the molecule. Due to the diverse range of functions of ncRNAs, their computational design and analysis has numerous applications in nano-technology, therapeutics, synthetic biology and materials engineering.

The RNA design problem is to find novel RNA sequences that are predicted to fold into target structure(s) while satisfying specific qualitative characteristics and constraints. RNA design can be modelled as a combinatorial optimization problem (COP) and is known to be computationally challenging or more precisely NP-hard. Numerous algorithms to solve the RNA design problem have been developed over the past two decades, however mostly ignore pseudoknots and therefore limit application to only a slice of real world modelling and design problems. Moreover, the few existing pseudoknot designer methods which were developed only recently, do not provide any evidence about the applicability of their proposed design methodology in biological contexts. The two objectives of this thesis are set to address these two shortcomings. First, we are interested in developing an efficient computational method for the design of RNA secondary structures *including* pseudoknots that show significantly improved *in-silico* quality characteristics than the state of the art. Second we are interested in showing the real-world worthiness of the proposed method by validating it experimentally. More precisely, our aim is to design instances of certain types of RNA enzymes (i.e. ribozymes) and demonstrate that they are functionally active. This would likely only happen if their predicted folding matched their actual folding in the *in-vitro* experiments.

In this thesis we present four contributions. First, we propose a novel *adaptive defect weighted sampling* algorithm to efficiently solve the RNA secondary structure design problem where pseudoknots are included. We compare the performance of our design algorithm with the state of the art and show that our method generates molecules that are thermodynamically more stable and less defective than those generated by state of the art methods. Moreover, we show when the effect of fitness evaluation is decoupled from the search and optimization process, our optimization method converges faster than the non-dominated sorting genetic algorithm (NSGA II) and the ant colony optimization (ACO) algorithm do. Second, we use our algorithmic development to implement an RNA design pipeline called `Enzymer` and make it available as an open source package useful for wet-lab practitioners and RNA bioinformaticians. `Enzymer` uses multiple sequence alignment (MSA) data to generate initial design templates for further optimization. Our design pipeline can then be used to reengineer naturally occurring RNA enzymes such as ribozymes and riboswitches. Our first and second contributions are published in the RNA section of the journal of Frontiers in Genetics. Third we use `Enzymer` to reengineer three different species of pseudoknotted ribozymes: a hammerhead ribozyme from the mouse gut metagenome, a hammerhead ribozyme from *Yarrowia lipolytica* and a glmS ribozyme from *Thermoanaerobacter tengcogensis*. We designed a total of 18 ribozyme sequences and showed the 16 of them were active *in-vitro*. Our experimental results have been submitted to the RNA journal and strongly suggest that `Enzymer` is a reliable tool to design pseudoknotted ncRNAs with desired secondary structure. Finally we propose a novel architecture for a new ribozyme based gene regulatory network where a hammerhead ribozyme modulates expression of a reporter gene when an external stimulus IPTG is present. Our *in-vivo* results show expected results in 7 out of 12 cases.

# Acknowledgments

As this chapter of my life is about to come to an end, I must say no single page of this document would have been existed if it was not because of the unconditional and continuous support I received from my father, mother and sister throughout my educational career. You filled my heart with hope; thank you for 35 years of love.

I would like to thank and offer the greatest of respect to my supervisors Dr. Nawwaf Kharma and Dr. Gregory Butler. While being patient, you raised the bar and kept pushing me. This would have not become possible if it was not because of Dr.Kharma's faith and Dr.Butler's patience. You helped me to aim for the right target. You shared with me the greatest of your advise while giving me the freedom to find my own path; thank you for your support and guidance.

When I was in search for external experts to challenge my work in an experimental setting, Dr. Jonathan Perreault from the RNA lab at the INRS Armand-Frappier Institute offered valuable help. Jonathan, you showed belief in my work and put the effort and resources to help me to show my work can go beyond the boring computer simulations. I thank you for 2 years of endless discussions and past midnight email communications. I would also like to thank Sabrine Najeh who is a Ph.D student of Jonathan for running the experimental protocols and generating wet-lab data.

I would like to thank the members of the Computational Structural Biology from the School of Computer Science at McGill university. In particular I would like to thank Dr. Jérôme Waldispühl, who taught me the first baby steps in field of structural bioinformatics and planted the seed of RNA engineering in my mind. Thank you for the hospitality over the year long duration I was your guest.

Words can not express how grateful I am for the support I received for over two years from Dr. Niles Pierce from the Pierce Lab at California Institute of Technology. I never met you but your countless in-depth, comprehensive, caring and helpful responses related to my work at odd hours of your local time filled my brain with inspiration and helped me to aim for the right target. Your love for what you do and the efforts you put to show me insights into what I was doing taught me how important it is to love what I do; thank you.

Halina Monkiewicz, as a graduate program advisor you took all the extra caring steps possible to help me to navigate my way around the department; thank you.

My friends and loved ones, those who spent significant amounts of time with me during this journey, you know who you are; thank you.

# Contents

# List of Figures

# List of Tables

# Glossary

***3'*** A key feature of all nucleic acids is that they have two distinctive ends: the 5' (5-prime) and 3' (3-prime) ends. This terminology refers to the 5' and 3' carbons on the sugar. For both DNA and RNA, the 5' end bears a phosphate, and the 3' end a hydroxyl group.. 3

***5'*** A key feature of all nucleic acids is that they have two distinctive ends: the 5' (5-prime) and 3' (3-prime) ends. This terminology refers to the 5' and 3' carbons on the sugar. For both DNA and RNA, the 5' end bears a phosphate, and the 3' end a hydroxyl group.. 3

***ACO*** ant colony optimization (ACO) is an optimization algorithm inspired by ant colonies efforts to collect resources for their colonies.. 8

***DNA promoter*** In genetics, a promoter is a region of DNA that initiates transcription of a particular gene. Promoters are located near the transcription start sites of genes, on the same strand and upstream on the DNA (towards the 5' region of the sense strand). Promoters can be about 100 to 1000 base pairs long.. 100

***DNA*** Deoxyribonucleic acid (DNA) is a molecule that carries the genetic instructions used in the growth, development, functioning and reproduction of all known living organisms and many viruses. DNA and ribonucleic acid (RNA) are nucleic acids; alongside proteins, lipids and complex carbohydrates (polysaccharides), they are one of the four major types of macromolecules that are essential for all known forms of life. Most DNA molecules consist of two biopolymer strands coiled around each other to form a double helix.. 2

***Enzyme*** Enzymes are macromolecular biological catalysts.. 4

***HHRz*** hammerhead ribozyme.. 91

***IPTG*** Isopropyl $\beta$-D-1-thiogalactopyranoside (IPTG) is a molecular biology reagent. This compound is a molecular mimic of allolactose, a lactose metabolite that triggers transcription of the lac operon, and it is therefore used to induce protein expression where the gene is under the control of the lac operator.. 9

***LacI*** The lac repressor (LacI) operates by a helix-turn-helix motif in its DNA binding domain binding base-specifically to the major groove of the operator region of the lac operon, with base contacts also made by residues of symmetry-related alpha helices, the "hinge" helices, which bind deeply in the minor groove. This DNA binding causes the specific affinity of RNA polymerase for the promoter sequence to increase sufficiently that it cannot escape the promoter region and enter elongation, and so prevents transcription of the mRNA coding for the Lac proteins. When lactose is present, allolactose binds to the lac repressor, causing an allosteric change in its shape. In its changed state, the lac repressor is unable to bind tightly to its cognate operator. This effect is referred to as induction, because it induces, rather than represses, expression of the metabolic genes. In vitro, Isopropyl IPTG is a commonly used allolactose mimic which can be used to induce transcription of genes being regulated by lac repressor.. 100

***NSGA II*** non dominant sorting genetic algorithm (NSGA II) is a multi objective optimization algorithm.. 8

***RFP*** red fluorescent protein.. 9

***RNA*** Ribonucleic acid (RNA) is a polymeric molecule essential in various biological roles in coding, decoding, regulation, and expression of genes. RNA and DNA are nucleic acids, and, along with lipids, proteins and carbohydrates, constitute the four major macromolecules essential for all known forms of life. Like DNA, RNA is assembled as a chain of nucleotides, but unlike DNA it is more often found in nature as a single-strand folded onto itself, rather than a paired double-strand.. 1

***Shine-Dalgarno*** The Shine-Dalgarno (SD) sequence is a ribosomal binding site in bacterial and archaeal messenger RNA, generally located around 8 bases upstream of the start codon AUG. The RNA sequence helps recruit the ribosome to the messenger RNA (mRNA) to initiate protein synthesis by aligning the ribosome with the start codon.. 94

***Thermoanaerobacter tengcongensis*** thermoanaerobacter is a genus in the phylum Firmicutes (Bacteria). Members of this genus are thermophilic and anaerobic, several of them were previously described as Clostridium species. *Thermoanaerobacter tengcongensis* is an anaerobic, saccharolytic, thermophilic bacterium isolated from a hot spring in Tengcong, China.. 8

***Yarrowia lipolytica*** Yarrowia is a fungal genus in the family Dipodascaceae. For a while the genus was monotypic, containing the single species Yarrowia lipolytica, a yeast that can use unusual carbon sources, such as hydrocarbons. This has made it of interest for use in industrial microbiology, especially for the production of specialty lipids. Molecular phylogenetics analysis has revealed several other species that have since been added to the genus. The yeast Yarrowia lipolytica presents specific physiological, metabolic and genomic characteristics, which differentiate it from the model yeast Saccharomyces cerevisiae.. 8

***cofactor*** A cofactor is a non-protein chemical compound or metallic ion that is required for a protein's biological activity to happen. These proteins are commonly enzymes, and cofactors can be considered “helper molecules” that assist in biochemical transformations.. 91

***gene expression*** Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product. These products are often proteins, but in non-protein coding genes such as transfer RNA (tRNA) or small nuclear RNA (snRNA) genes, the product is a functional RNA.. 4

***glmS*** The Glucosamine-6-phosphate activated ribozyme (glmS ribozyme) is an RNA structure that is both a ribozyme, since it catalyzes a chemical reaction, and a riboswitch, since it regulates genes in response to concentrations of a metabolite. It was originally identified using bioinformatics in the 5' untranslated regions of glmS genes. The GlmS enzyme catalyzes the production of glucosamine-6-phosphate (GlcN6P), and the glmS ribozyme is dependent on GlcN6P to achieve catalysis of its own cleavage.. 8

***hammerhead ribozyme*** The hammerhead ribozyme is a RNA molecule motif that catalyzes reversible cleavage and joining reactions at a specific site within an RNA molecule. It serves as a model system for research on the structure and properties of RNA, and is used for targeted RNA cleavage experiments, some with proposed therapeutic applications.. 8

***in-silico*** is an expression used to mean "performed on computer or via computer simulation".. 6

***in-vitro*** studies are performed with microorganisms, cells, or biological molecules outside their normal biological context. Colloquially called "test-tube experiments", these studies in biology and its sub-disciplines have traditionally been done in test tubes, flasks, Petri dishes, etc.. 1

***in-vivo*** those in which the effects of various biological entities are tested on whole, living organisms, usually animals, including humans, and plants. 6

***metabolite*** Metabolites are the intermediates and products of metabolism. The term metabolite is usually restricted to small molecules. Metabolites have various functions, including fuel, structure, signalling, stimulatory and inhibitory effects on enzymes, catalytic activity of their own (usually as a cofactor to an enzyme), defence, and interactions with other organisms (e.g. pigments, odourants, and pheromones).. 91

***multiple sequence alignment (MSA)*** A multiple sequence alignment (MSA) is a sequence alignment of three or more biological sequences, generally protein, DNA, or RNA. In many cases, the input set of query sequences are assumed to have an evolutionary relationship by which they share a linkage and are descended from a common ancestor. From the resulting MSA, sequence homology can be inferred and phylogenetic analysis can be conducted to assess the sequences' shared evolutionary origins.. 1

***ncRNA*** A non-coding RNA (ncRNA) is a functional RNA molecule that is transcribed from DNA but not translated into proteins. Epigenetic related ncRNAs include miRNA, siRNA, piRNA and lncRNA. In general, ncRNAs function to regulate gene expression at the transcriptional and post-transcriptional level.. 4

***nucleic acid*** are biopolymers, or large biomolecules, essential for all known forms of life. Nucleic acids, which include DNA (deoxyribonucleic acid) and RNA (ribonucleic acid), are made from monomers known as nucleotides. Each nucleotide has three components: a 5-carbon sugar, a phosphate group, and a nitrogenous base. If the sugar is deoxyribose, the polymer is DNA. If the sugar is ribose, the polymer is RNA. When all three components are combined, they form a nucleotide. Nucleotides are also known as phosphate nucleotides. 1

***plasmid*** is a small DNA molecule within a cell that is physically separated from a chromosomal DNA and can replicate independently. Plasmids are considered replicons, a unit of DNA capable of replicating autonomously within a suitable host. However, plasmids, like viruses, are not generally classified as life.. 96

***quantum electrodynamics*** in particle physics, quantum electrodynamics (QED) is the relativistic quantum field theory of electrodynamics. In essence, it describes how light and matter interact and is the first theory where full agreement between quantum mechanics and special relativity is achieved. QED mathematically describes all phenomena involving electrically charged particles interacting by means of exchange of photons and represents the quantum counterpart of classical electromagnetism giving a complete account of matter and light interaction. 1

***ribosome*** The ribosome is a complex molecule made of ribosomal RNA molecules and proteins that form a factory for protein synthesis in cells.. 14

***riboswitches*** a riboswitch is a regulatory segment of a messenger RNA molecule that binds a small molecule, resulting in a change in production of the proteins encoded by the mRNA. Thus, a mRNA that contains a riboswitch is directly involved in regulating its own activity, in response to the concentrations of its effector molecule.. 5

***uncertainty*** in quantum mechanics, the uncertainty principle, also known as Heisenberg's uncertainty principle, is any of a variety of mathematical inequalities asserting a fundamental limit to the precision with which certain pairs of physical properties of a particle, known as complementary variables, such as position x and momentum p, can be known. 1

***upstream gene*** In molecular biology and genetics, upstream and downstream both refer to relative positions in DNA or RNA. Each strand of DNA or RNA has a 5' end and a 3' end, so named for the carbon position on the deoxyribose (or ribose) ring. By convention, upstream and downstream relate to the 5' to 3' direction in which RNA transcription takes place. Upstream is toward the 5' end of the RNA molecule and downstream is toward the 3' end. When considering double-stranded DNA, upstream is toward the 5' end of the coding strand for the gene in question and downstream is toward the 3' end. Due to the anti-parallel nature of DNA, this means the 3' end of the template strand is upstream of the gene and the 5' end is downstream.. 90

***wave-particle duality*** wave-particle duality is the concept that every elementary particle or quantic entity may be partly described in terms not only of particles, but also of waves. It expresses the inability of the classical concepts "particle" or "wave" to fully describe the behaviour of quantum-scale objects. 1

# Chapter 1

# Introduction

## 1.1 Preface

The first section of this chapter gives an overview of the study of *nucleic acid*s, in particular ribonucleic acid (RNA) nanotechnology and its applications in materials engineering, synthetic biology and nano medicine. We describe the key discipline areas of RNA nano engineering and highlight the importance of RNA structure design and analysis. In the second section, we present the main objective of this thesis as the development of enhanced computational methods for the design of functional *RNA* structures. In particular, we are interested in improving on our current abilities to engineer novel functional RNAs and also to verify the applicability of our methods in biological contexts. Next we describe a summary of the three contributions which lead us to meet our objectives. First we introduce a new algorithm for the design of RNAs with targeted secondary structures including a complex but important structural feature, the pseudoknot. Our algorithm utilizes a novel Boltzmann sampling technique to efficiently solve the RNA design problem. Our second contribution is a software named `Enzymer` which implements a complete design pipeline to reengineer naturally occurring and functional RNAs. Enzymer leverages our algorithmic development as well as the evolutionary information obtained from *multiple sequence alignment (MSA)* analysis to design RNA enzymes. Our third contribution is to demonstrate the RNAs designed by `Enzymer` deliver the expected function *in-vitro*. We end the chapter by presenting the outline of this thesis document.

## 1.2 Nucleic acids and nanotechnology

The term *"nanotechnology"* reminds me of Niels Bohr's famous saying: *"Everything we call real is made of things that can not be regarded as real"*. Nanotechnology is the engineering of functional systems at incredibly small scales where physical phenomena such as inertia and gravity vanish from view; instead the interactions of matter and energy are dominated by other phenomena such as *quantum electrodynamics*, *wave-particle duality* and *uncertainty*. The field of nanotechnology

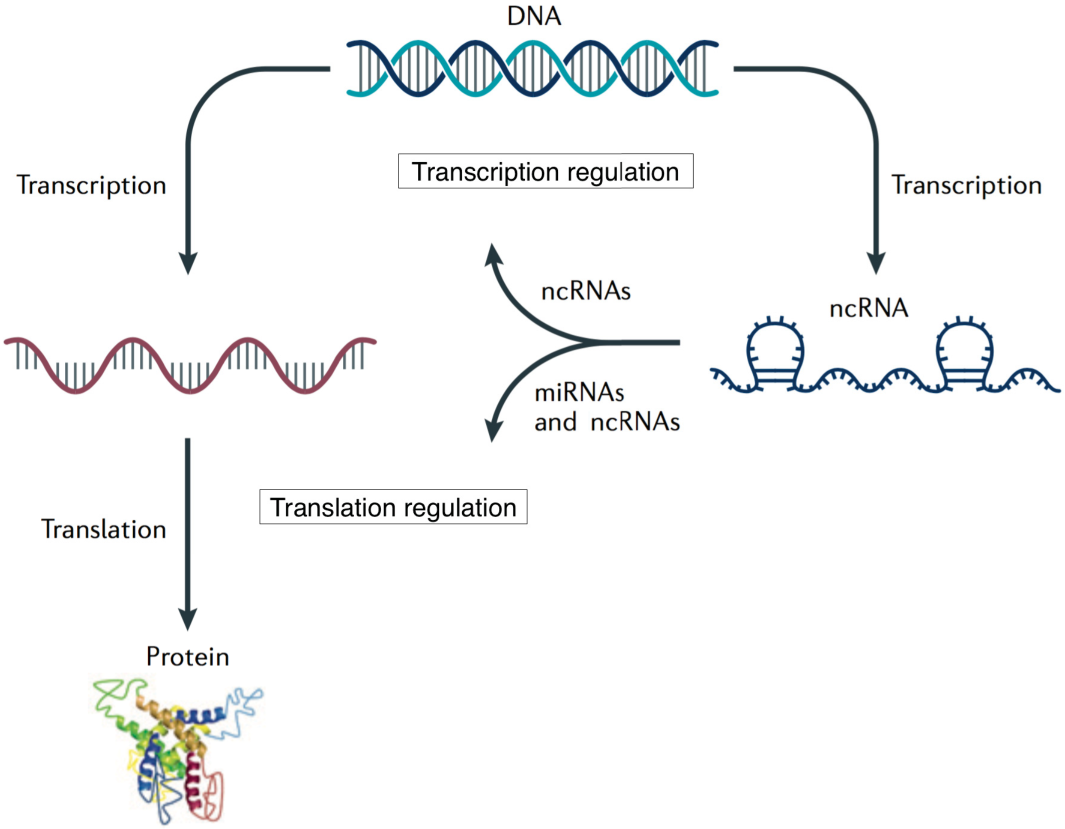


Figure 1: Central dogma of molecular biology. Via a process named *transcription*, the genetic information flows from DNA to two different types of RNA molecules: coding or messenger RNA (mRNA) and non-coding RNA (ncRNA)(Wahlestedt, 2013). The mRNA molecules will then be translated into proteins via a process named *translation* and ncRNAs will take different key regulatory roles.

taking advantage of nucleic acids has its origin in the works of Nadrian Seeman and coworkers (Winfree et al., 1998) with the focus on the development of deoxyribonucleic acid (*DNA*) nano-objects (Garibotti et al., 2007; Seeman, 2010). On the other hand Eric Westhof, Nocles Leontis, Luc Jaeger, Piexuan Guo and Bruce Shapiro pioneered the use of ribonucleic acid (RNA) molecules to engineer functional nano particles with the aim of tackling obstacles in nanomedicine, synthetic biology and material engineering (Grabow et al., 2012a; Guo, 2010; Leontis et al., 2006; Shapiro et al., 2008). The central dogma of molecular biology illustrated in Figure 1 reflects the fundamental role the DNA and RNA molecules play in all forms of life and Figure 2 gives a closer look at the bases that constitute these two polymer molecules.

DNA nanotechnology uses the nature of DNA complementarity to construct objects by formation of canonical Watson-Crick (A-T and G-C) pairs between the four different bases Adenine, Guanine, Cytosine and Uracil as illustrated by Figure 2. Formation of canonical base pairs provides the possibility of engineering numerous DNA 3D nano-scaffolds with different connectivities

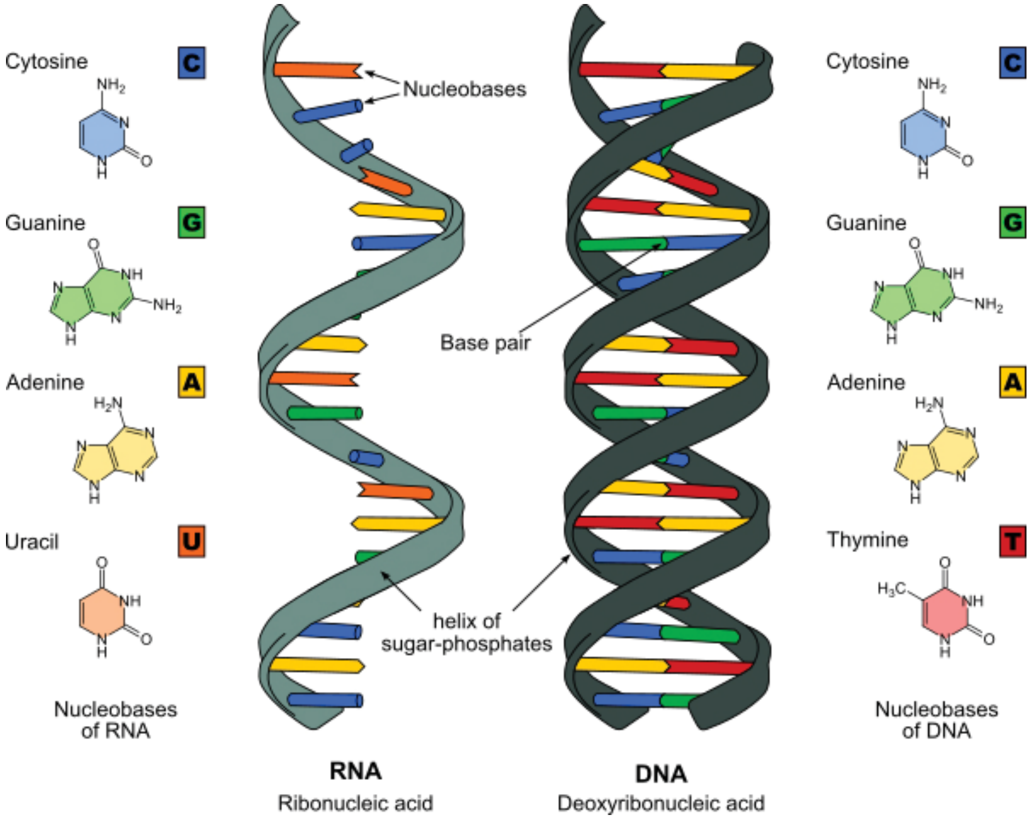


Figure 2: DNA and RNA polymers. DNA (right) and RNA (left) are both polymers of nucleotides. The DNA is composed of four different nucleic acid bases namely **C**ytosine (C), **G**uanine (G), **A**denine (A) and **T**hymine (T). In RNA, **T**hymine bases are replaced by **U**racil (U). The sugar backbone of the DNA polymer is made of deoxyribose and the sugar backbone of the RNA polymer is made of ribose. Generally speaking, the DNA polymer is found to be double stranded while the RNA polymer is found to be single stranded with the exception of some viruses with carry double stranded RNAs (Weber et al., 2006). Reading from top to bottom (or from *5'* to *3'* direction), the above RNA sequence can be read as CGAU.

and structural features (Andersen et al., 2008; Goodman et al., 2008; Yang, 2015). DNA nano structures have been fabricated to function as DNA nano-capsules for targeted delivery of drugs and other molecules (Mora-Huertas et al., 2010; Sun et al., 2014; Yang et al., 2009), or to build functional nanoboxes (Aherne et al., 2010), DNA origami (Marras et al., 2015) and nano-robots (Elbaz and Willner, 2012; Fu and Yan, 2012).

In spite of DNA nanostructures demonstrating the potential to develop programmable nano scaffolds (Jones et al., 2015), DNA polymers are often not able to mimic the diverse biological functions of RNAs. Due to unique structural, chemical and physical properties of the RNAs which have some advantages compared to those of DNA (Guo, 2010), RNA molecules can serve as an attractive biochemical material for applications in synthetic biology (Chappell et al., 2015; Ruder et al., 2011) and fabrication of functional nonostructures (Geary et al., 2014). RNAs have the ability to form non-canonical base pairs (Lemieux and Major, 2002) leading to the natural library of diverse structural motifs (Hendrix et al., 2005) which in turn create a wide array of complex structures many of which posses functional properties similar to proteins. Notably the ensemble of RNA structures offers even more diversity compared to that of proteins as RNAs have 7 degrees of freedom in their polymer backbone (Richardson et al., 2008) while proteins have only 4 (Richardson, 1981). Furthermore, RNAs can be integrated into native cells and take advantage of expression within the cells to perform different functions (Brophy and Voigt, 2014; Delebecque et al., 2011; Lienert et al., 2014) and therefore potentially act as effective therapeutic agents (Hong and Nam, 2014; Kole et al., 2012).

RNAs with catalytic activities or functional RNAs are termed non-coding RNAs (*ncRNA*s) as they perform their functionality directly and not via their protein products (Mattick and Makunin, 2006). Due to their diverse range of functionalities, ncRNA have been used to build synthetic genetic platforms that make it possible to specifically target and silence other RNAs with the aim of regulating *gene expression* (Afonin et al., 2008b; Chen et al., 2010; Isaacs et al., 2006; Khalil and Collins, 2010; Kharma et al., 2016), to develop influenza virus vaccines (Mueller et al., 2010), to developing RNA *Enzyme*s targeting the HIV RNA (Scarborough et al., 2014), or building synthetic genetic switches (Findeiss et al., 2015; Lucks et al., 2011; Xie et al., 2011).

### 1.2.1 Foundations of RNA nanotechnology

The relationship between RNA polymer sequence and RNA structure plays a fundamental role in characterizing the functions of RNA structures (Mortimer et al., 2014). Figure 3 shows the key areas which define the foundations of RNA nanotechnology that are related to the understanding of the relationship between RNA sequence, RNA structure and RNA function. These key areas together define a framework for determining the RNA structure-function relation, as well as designing novel RNA polymer sequences required to build ncRNA structures with desired function.

When a natural or artificial RNA polymer sequence is provided, the first issue is to determine the secondary structure (2D), then the three dimensional (3D) structure and finally to characterize the function. This process is called structure-function determination which is either accomplished by experimental approaches such as X-ray crystallography (Ban et al., 1998; Jackson et al., 2015) or nuclear magnetic resonance (NMR) (Feigon, 2015; Varani and Tinoco, 1991) or in part by computational methods (Chang et al., 2013; Mortimer et al., 2014) followed by wet-lab experimentations (Frommer et al., 2015). RNA secondary structure provides the scaffold of the 3D structure and therefore, whether derived from experimental or computational approaches, provides extremely helpful information in determining the 3D structure as well as the function (Dieterich and Stadler, 2013; Leontis and Westhof, 2003). Once the 3D structure is determined it becomes possible to study further the relationship between the sequence, structure and function of the molecule. On the other hand, RNA structure design first starts with a having target 2D or 3D structure as a design template. If starting from the 3D structure, then a set of compatible 2D structures are inferred from it and the final issue is to derive sequences that are predicted to fold into the desired

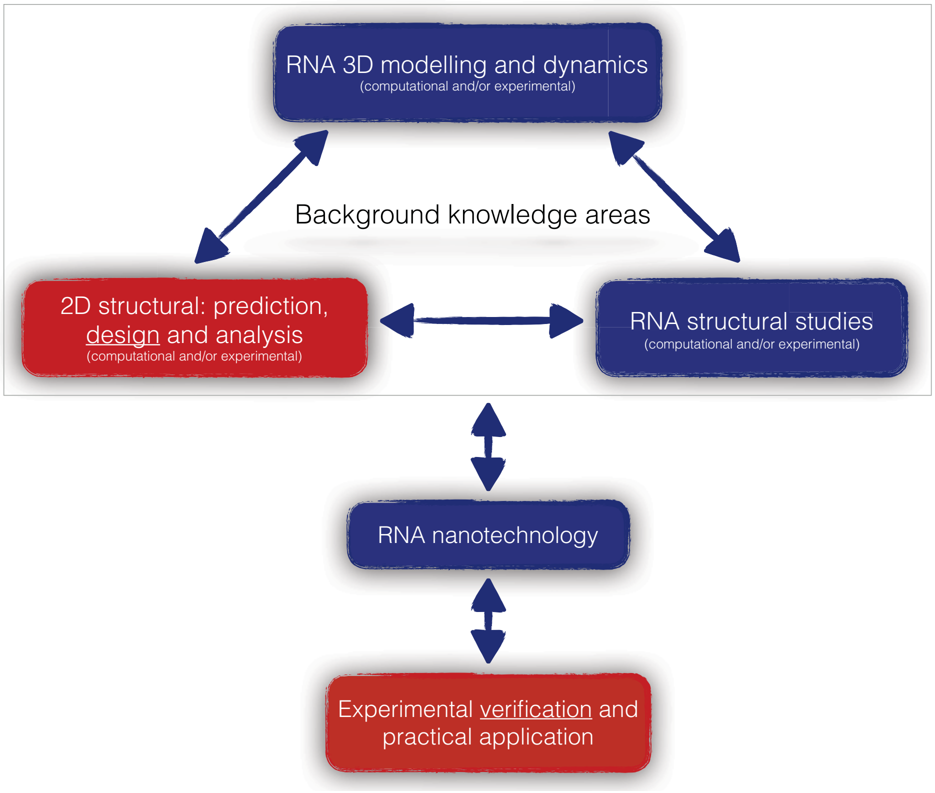

RNA 3D modelling and dynamics
(computational and/or experimental)
Background knowledge areas
2D structural: prediction, design and analysis
(computational and/or experimental)
RNA structural studies
(computational and/or experimental)
RNA nanotechnology
Experimental verification and practical application

Figure 3: components of RNA nanotechnology. The red boxes represent the areas of contribution of this thesis: 2D design and experimental verification.

2D and eventually 3D structure, hence delivering the expected functionality. Figure 3 shows how the three key steps of RNA structural studies; 3D modelling, and 2D structure prediction and design, constitute the foundation of RNA nanotechnology.

From the prospective of design, the modular components of RNA nano technology can be semantically divided into two groups: functional and tectonic (Figure 4). The functional group represents ncRNAs with desired functionality such as the ones found in nature playing important roles in regulating key cellular processes including short interfering RNAs (siRNA) (Novina et al., 2002; Reynolds et al., 2004), ribozymes (Pley et al., 1994; Roth et al., 2014), aptamers (Germer et al., 2013; Zhou et al., 2015), *riboswitches* (Breaker, 2012; Serganov and Nudler, 2013) and micro RNAs (miRNAs) (Ambros, 2004; Gurtan and Sharp, 2013). The tectonic group can be further split into structural motifs (Figure 4, bottom right) and interacting motifs (Figure 4, bottom left). Helix (Dock-Bregeon et al., 1989), three or four way junction (Hohng et al., 2004; Lescoute and Westhof, 2006), pseudoknot (Brierley et al., 2007; Staple and Butcher, 2005), k-tun (Klein et al., 2001), RA-motif (Grabow et al., 2012b) and nono-corner (Dibrov et al., 2011) are examples of structural motifs. Sticky ends (Roy and Ibba, 2006), paranemic motifs (Afonin et al., 2008a) and kissing loops

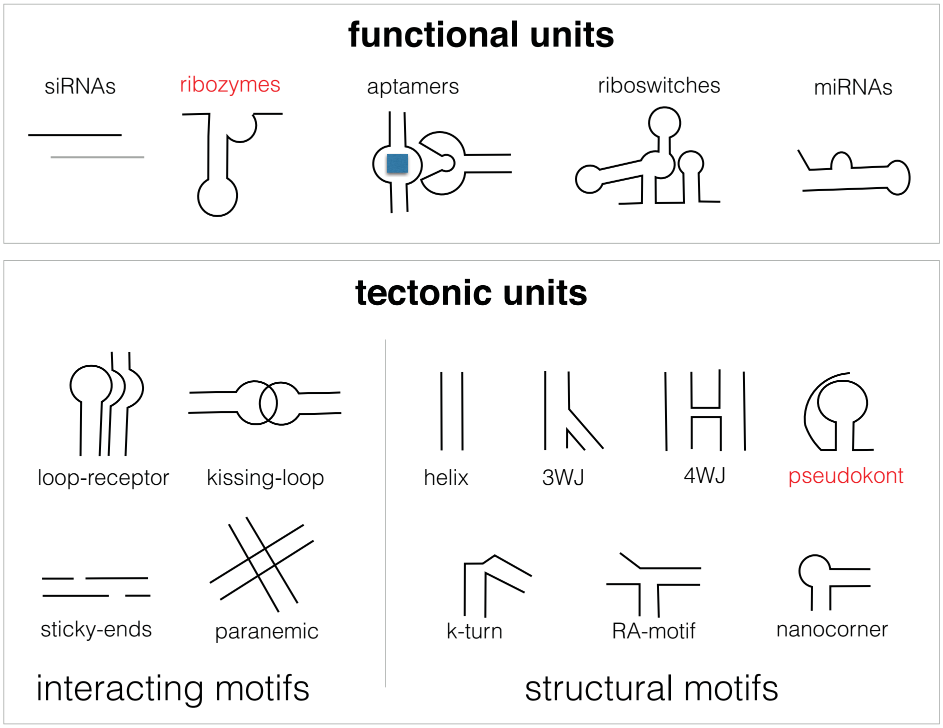


Figure 4: Modular components of RNA. Showing the two semantic classes of RNA nanotechnology namely "functional units" (top) and "tectonic units" (bottom) in their secondary structure representations. Computational design and experimental validation of ribozymes (marked red under the functional units) that contain pseudoknot motifs (marked red under the tectonic units) are another area of contribution of this thesis.

(Cao and Chen, 2011) are examples of interacting motifs. By rationally combining the tectonic and functional units, 2D scaffolds and 3D modules with predicted structure and desired function can be designed *in-silico* and then tested *in-vivo* and *in-vitro*.

### 1.2.2 Applications of RNA secondary structure design

As illustrated in Figure 3, RNA secondary structure plays a key role in both prediction (termed as RNA folding) and also design (termed as inverse folding) of RNA 3D structure and function. The inverse RNA folding or the problem of designing RNA sequences that fold into targeted secondary structure, was first introduced in the early 1990's in Vienna (Hofacker et al., 1994) and implemented as a software named RNAinverse (Hofacker, 2003). The RNA design objective is to find a set of possible RNA polymer sequences which can be predicted to have a targeted set of folding attributes as predicted by structure prediction algorithms or verified by experimental methods such as X-ray or NMR. RNA inverse folding is useful in designing novel RNAs with desired function, to re-engineer naturally occurring RNA enzymes or to build artificial genetic circuits capable of regulating cellular functions. Notably RNA sequences generated by inverse folding approaches, have recently been used as seeds for sequence based genomic search methods (Retwitzer et al.,

2015) and have been shown to lead to the discovery of novel naturally occurring functional RNAs which had remained unnoticed in previous multiple sequence alignment and phylogenetic studies (Ruzzo and Gorodkin, 2014).

## 1.3 Thesis overview

### 1.3.1 Thesis objectives

Computational methods for the design of RNA secondary structures with targeted attributes have been extensively studied and developed over the past decade. The RNA inverse folding problem which in almost all cases has been defined as a combinatorial optimization problem (COP) is computationally complex (Haleš et al., 2015; Schnall-Levin et al., 2008). A diverse range of methods utilizing different combinatorial optimization methods such as simulated annealing, adaptive walk, Boltzmann sampling, graph decomposition, ant colony optimization and genetic algorithms have been developed to solve the inverse folding problem. Despite the wealth of subsequent RNA inverse folding methods which emerged after RNAinverse, there is no single method that can suit all possible use cases. For instance most of the existing methods ignore an important structural motif named pseudoknot and therefore limit their use. Furthermore, the few existing methods that can handle pseudoknots (Gao et al., 2010; Kleinkauf et al., 2015; Taneda, 2011) only present empirical results obtained from *in-silico* simulations and do not provide any evidence to support the applicability of their computational methods in any biological context. Consequently, there is a lack for a full inverse folding pipeline to bridge the gap between the computational design and experimental verification of pseudoknotted ncRNAs that can potentially be used as a way to regulate expression of targeted gene products such as the experiments done by (Dotu et al., 2014; Kharma et al., 2016). The objective of this thesis is set to address these shortcomings. The first objective is to develop enhanced computational algorithms for the design of functional RNAs that include pseudoknots enabling one to design more complex structures than what is currently possible. Second, we are interested in defining a complete inverse folding pipeline by leveraging our algorithmic developments as well as the evolutionary information obtained from the MSA data available in RNA homology repositories such as `RFam` (Gardner et al., 2009), and using the pipeline to reengineer naturally occurring RNA enzymes with desired attributes. Third, we are interested in verifying the applicability of our computational design methodology by designing different species of ribozymes and then verifying their functionality *in-vitro*. Fourth, we are aiming to explore the idea of using ribozymes to implement novel artificial genetic networks capable of modulating the expression of genes that are embedded in the network.

### 1.3.2 Thesis contributions

**An adaptive weighted sampling algorithm (Enzymer):** We develop a novel and efficient *"adaptive defect weighted sampling algorithm"* (Algorithm 5) (Zandi et al., 2016) for designing

RNA secondary structures including pseudoknots. We name our method Enzymer. We use a non-redundant dataset named Pseudobase (Van Batenburg et al., 2000) composed of naturally occurring and experimentally characterized ncRNAs, to benchmark the performance of our algorithm. We compare our results with the results we obtained from the state of the art and show our method succeeds more often, and the RNAs it generates are predicted to be thermodynamically more stable (Figure 19) and less defective (Figure 15, 18) than those generated by the others. We also show our adaptive defect weighted sampling method is a more efficient combinatorial optimization strategy than two of the most successful combinatorial optimization strategies namely the non-dominant sorting genetic algorithm II (*NSGA II*) (Deb et al., 2000) and the ant colony optimization (*ACO*) (Dorigo et al., 2006) adopted by the other related programs. To characterize the efficiency of our search and optimization strategy, we decouple the effect of fitness evaluation from the search and optimization procedures and show our adaptive defect weighted sampling strategy leads to faster convergence when compared with NSGA II and ACO. We show that the faster convergence rate of our method is because of the smaller number of fitness evaluations it requires (Figure 22) compared to NSGA II and ACO when at the same time leading to higher quality solutions. Our results imply that regardless of the optimization context (i.e optimizing on fitness of RNA sequences) our proposed method converges faster than NSGA II and ACO do, and can generate higher quality solutions (Zandi et al., 2016). In sections 5.5.3 and 5.6.3 we provide in-depth analysis to support our claims regarding originality and significance.

**Enzymer pipeline:** We present a complete RNA design pipeline named Enzymer-pipeline which uses our novel combinatorial optimization strategy. The Enzymer-pipeline (Algorithm 9) utilizes the evolutionary sequence data which can be obtained from public homology libraries such as Rfam or other multiple sequence alignment methods to extract the homology profile of the catalytic core of naturally occurring ribozymes (i.e. RNA enzymes). The Enzymer-pipeline utilizes the extracted homology profile to generate design templates which will be used as initial seeds for our weighted sampling algorithm. We use our pipeline to reengineer naturally occurring ribozymes. The outcome of this contribution is a python 2.7 software, which can easily be used by wet-lab practitioners to reengineer naturally occurring RNAs or to design new artificial ones. We published the Enzymer pipeline in the journal of Frontiers in Genetics (impact factor 3.78) (Zandi et al., 2016).

**Experimental validation:** We bridge the gap between pseudoknotted ncRNA design and experimental validation of the functionality of the designed RNAs. We obtain the consensus secondary structure of three self-cleaving *cis-acting* pseudoknotted ribozymes: a *hammerhead ribozyme* from the mouse gut meta-genome, a hammerhead ribozyme from the fungus *Yarrowia lipolytica*, and a *glmS* ribozyme from *Thermoanaerobacter tengcongensis*. We generate a total of 18 ribozyme sequences and test them *in-vitro* and show 16 of them are active (Figures 24, 25, 26). These results I) show Enzymer is a reliable tool for designing active ribozymes including pseudoknots, an accomplishment no other program has demonstrated before and, II) show for the first time that the underlying energy model of Dirks and Pierce (Dirks and Pierce, 2003) can successfully capture the structural properties required to design functional pseudoknotted ribozymes. Notably the glmS ribozyme we designed is a relatively large molecule with complex structural attributes. To the best of our knowledge this is the first complete study of inverse folding and validation of pseudoknotted ribozymes. We have submitted the experimental data to the RNA Journal (impact factor 4.94) (Zandi et al., 2018).

**New synthetic gene regulatory network:** We propose a new ribozyme-based gene regulatory network architecture based on activation of a hammerhead ribozyme which can be triggered by an external stimuli, *IPTG*. Then we re-engineer a naturally occurring *cis-acting* pseudoknotted hammerhead ribozyme from *Yarrowia lipolytica* (Barth and Gaillardin, 1997) and embed the designed sequence in the proposed architecture. We use the final construct to modulate expression of a downstream reporter gene *in-vivo*, which in our case is the Red Fluorescent Protein (*RFP*). In 7 out of 12 test cases we observed when the system was triggered by addition of IPTG, the ribozymes repressed expression of the RFP gene (Figure 27).

### 1.3.3 Thesis outline

**Chapter 2:** We present a comprehensive review of RNA structure and algorithmics. We describe the mathematical notation used to characterize key molecular and structural properties of RNA secondary structures.

**Chapter 3:** RNA design is a specific class of combinatorial optimization problems (COPs). In this chapter we present the formal description of COPs and the computational complexity associated with them. We present a thorough review of the different classes of COPs as well as the existing methods to solve them.

**Chapter 4:** We present a literature review related to combinatorial optimization methods used in the context of RNA secondary structure design. We provide an in depth review of the state of the art in the field of RNA inverse folding and highlight some of the shortcomings of the existing methods with emphasis on pseudoknots and experimental verification.

**Chapter 5:** We present the complete `Enzymer` pipeline which combine our first adaptive defect weighted sampling algorithm) and second (design pipeline) contributions. We present the *in-silico* experimental results generated by `Enzymer` and compare the results with the results generated by the state of the art software.

**Chapter 6:** We present our third and fourth contributions. We use `Enzymer` to design pseudoknotted ribozymes. We present the wet-lab experimental protocols and results of *in-vitro* verification for three different ribozyme species. We also present our proposed architecture for a synthetic gene regulatory network which combines a hammerhead ribozyme with the coding sequence of a reporter green and present the *in-vivo* experimental results.

**Chapter 7:** We present a summary of our studies and contributions, discuss the current limitations and propose areas for future work and exploration.

# Chapter 2

# Foundations of RNA Computational Biology

## 2.1 Preface

In this chapter we present the foundations of RNA sequence-structure mapping. We describe the RNA folding process based on the set of experimentally measured energy parameters of Turner (Mathews et al., 1999) including the energy parameters related to the formation of pseudoknots as measured by Dirks and Pierce (Dirks and Pierce, 2003). We present the state of the art in computational methods for single sequence structure prediction. We describe how the conserved evolutionary information that can be obtained from multiple sequence alignment methods can be used to improve the quality of structure prediction.

## 2.2 RNA sequence and structure

### 2.2.1 RNA sequence

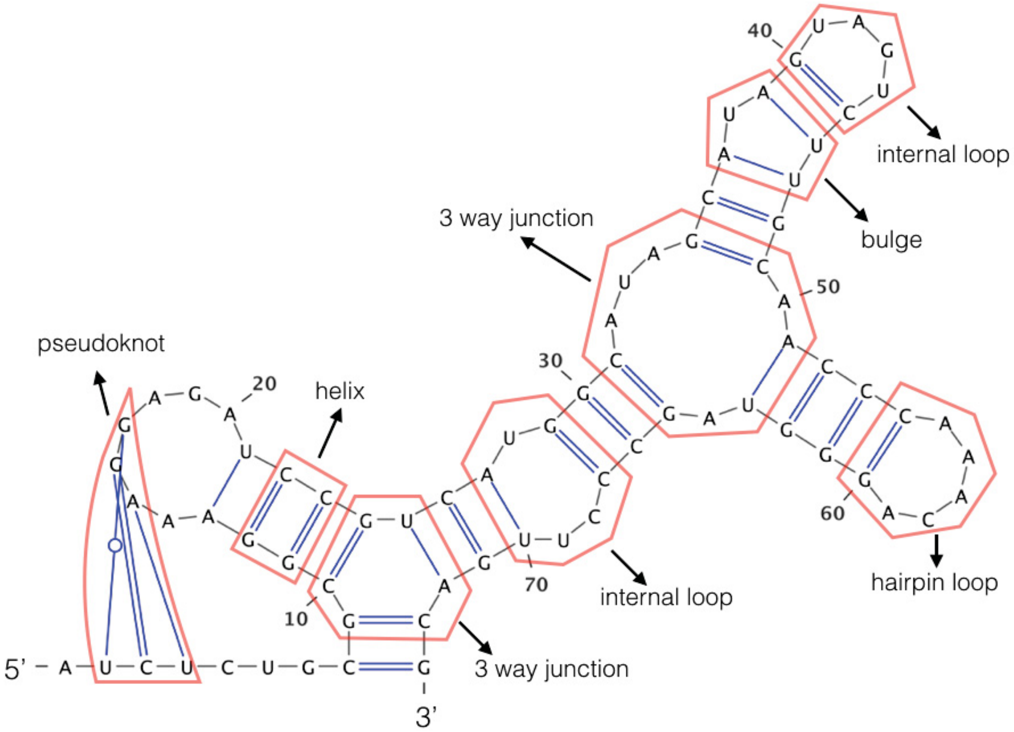
We denote an RNA polymer composed of $n$ nucleotides by sequence $\phi = \phi_1 ... \phi_n$ where $\phi_i \in \{A, U, G, C\}$ for $i = 1, ..., n$. The alphabets correspond to the four different nucleotide bases that compose the RNA polymer and stand for **A**denine, **U**racil, **G**uanine and **C**ytosine respectively. The nucleotide $\phi_i$ is chained to $\phi_{i+1}$ via phosphate bonding such that the $5'$ carbon atom of the sugar backbone of $\phi_i$ is connected to the $3'$ carbon atom of the sugar backbone of $\phi_{i+1}$ through phospha te bonds. The first nucleotide in the polymer chain ($\phi_1$) defines the $5'$ end and the last nucleotide ($\phi_n$) defines the $3'$ end.

In an RNA polymer each nucleotide can form base pair with other nucleotides. Watson-Crick $\{(A-U), (U-A), (G-C), (C-G)\}$ base pairs and Wobble $\{(G-U), (U-G)\}$ base pairs are termed canonical base pairs and account for approximately 60% of the base pairs in an RNA polymer. The great majority of the remaining bases participate in some other kind of edge-to-edge interactions with one or more other bases (Leontis and Westhof, 2001). However the formation of canonical base pairs are energetically more favourable and have great impact the global fold of the RNA sequence (Stombaugh et al., 2009). Therefore it is a reasonable simplification to only take the canonical base-pairs into consideration when we model and analyze RNA secondary structures (Lemieux and Major, 2002; Martinez, 1984; Wyatt et al., 1989).

### 2.2.2 RNA secondary structure

The flexibility of the the sugar phosphate backbone of an RNA polymer allows for formation of base pairs via hydrogen bonding between any bases that are at least 3 nucleotides apart. Formation of canonical base pairs bends the polymer and folds it into its secondary structure. A Secondary structure $\tau$ can be specified by a set of base pairs $(\phi_i, \phi_j)$ where $1 \leq i \leq j \leq n$ such that positions $i$ and $j$ are paired, $i+3 \leq j$, and $(\phi_i, \phi_j) \in \{(A-U), (G-C), (G-U), (U-A), (C-G), (U-G)\}$. Each position can be involved in exactly one canonical base pair. The base pairs of a secondary structure describe the base pairing interactions formed by hydrogen bonding in a corresponding tertiary structure. For two base pairs $(\phi_i, \phi_j)$ and $(\phi_k, \phi_l)$, a non-nested loop or a pseudoknot forms if either of the nesting rules $i \leq k \leq l \leq j$ or $k \leq i \leq j \leq l$ is violated. Characterizing the base pairing energy of non-nested base pairs is experimentally difficult and therefore the classical secondary structure prediction algorithms ignore them. Pseudoknots are abundant in nature and exist in specific types (Condon et al., 2004) and are known to play key roles in functionality of active RNAs (Staple and Butcher, 2005). RNA secondary structures can be expressed by the alphabet $S = \{(, [, ., ], )\}$ where an opening parenthesis represents the first base or the opening base of a nested base pair, an opening bracket represents the opening base in a pseudoknot, a dot represents an unpaired or a single base and the closing parenthesis and brackets represent the closing bases of nested and non-nested base pairs, respectively.

When an RNA folds into a secondary structure, a set of Secondary Structure Elements (SSE) emerge. Figure 5 shows an RNA secondary structure and annotates the different types of SSEs. The SSEs are *hairpin*, *internal loop*, *bulge loop*, *k-way junction* and *helix*. A hairpin loop is formed when an RNA strand folds back on itself. In an internal loop, at least one base is unpaired on each strand of the loop separating two paired regions. A bulge has unpaired nucleotides on only one strand where the other strand has uninterrupted base pairing. A k-way loop occurs when double-stranded regions separated by any number of unpaired nucleotides, come together. A helix or stack forms when perfect one on one pairing is formed between two regions of the strand. Pseudoknots are often called tertiary motifs and are formed when unpaired regions of a loop form base pairs with other bases outside of that loop. Depending on the nucleotide composition of the subsequence that is involved in the formation of each type of SSE, a specific amount of energy that can be approximated by nearest-neighbour energy models is released. The sum of the energy values released by formation of the SSEs is known as the free energy of sequence $\phi$ folding into structure $\tau$. We denote the free energy of sequence $\phi$ folding into secondary structure $\tau$ by $\Delta G(\phi, \tau)$. It has



.[[[...((((((.]]]...))))(((.(((...(((.((....))))).((((.....)))).)))..)))))

Figure 5: The secondary structure of an RNA sequence from its 5’ end to 3’ end and the corresponding dot bracket notation. The secondary structure elements (SSEs) are annotated. The blue lines represent the hydrogen bonds between two bases. There are two hydrogen bonds between G and C, one hydrogen bond between A and U, and one hydrogen bond between G and U bases.

been shown that RNA sequences tend to fold into secondary structures by lowering their folding energy (Mathews and Turner, 2006). The structure with lowest energy is called the minimum free energy (MFE) structure.

### 2.2.3 RNA tertiary structure

In the next level of organization, the tertiary structure, the secondary structure elements are associated through numerous van der Waals contacts, specific hydrogen bonds via the formation of a small number of additional Watson-Crick pairs and/or unusual pairs involving hairpin loops or internal bulges. RNA tertiary structure comprises those interactions involving (a) two helices, (b) two unpaired regions, or (c) one unpaired region and a double-stranded helix. The interactions between two helices are basically of two types: either two helices with a contiguous strand stack on each other, or two distant helices position themselves so that their shallow grooves fit. An unpaired region belongs to either a single-stranded stretch (forming an internal loop or a bulge) or a hairpin loop closing a helix. Interactions between two unpaired regions lead to pseudoknots if a single loop is involved and to loop-loop motifs otherwise. Interactions between an unpaired region and a double-stranded helix can lead to various types of motifs. Pairing of a single-stranded

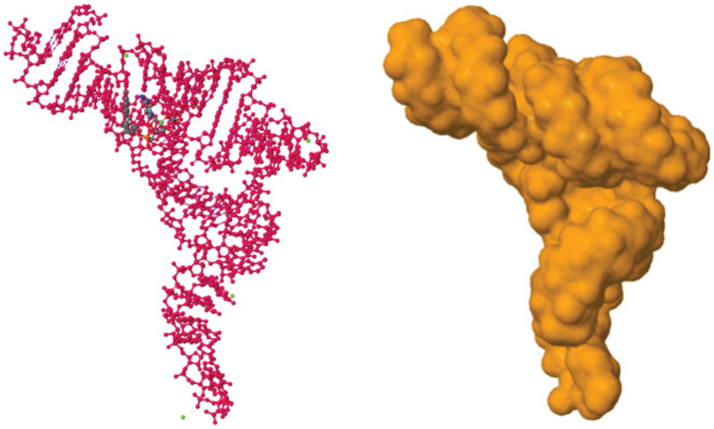


Figure 6: The hammerhead ribozyme is a self-cleaving RNA broadly dispersed across all kingdoms of life. The relative positions of individual atoms (left) and the space filling model based on the solved crystallized 3D structure (right) from the Protein Data Bank (PDB) with ID=5DH6 (Rose et al., 2013) are shown.

stretch, either in the deep or the narrow groove of a double helix, yields a triple helix. We refer the reader to a comprehensive description and classification of 3D RNA motifs given by (Leontis and Westhof, 2001). RNA 3D structure is the complete positional specification of all of the atoms in space resulting from formation of the tertiary structure. Figure 6 shows the 3D structure of an RNA enzyme called hammerhead ribozyme (Afonin et al., 2008a).

### 2.2.4 RNA quaternary structure

RNA quaternary structure refers to the interaction between one RNA and separate nucleic acid molecules or between an RNA and proteins. RNAs often require the formation of molecular complexes via binding and interaction with proteins to express their function.

Several functional RNAs such as the glmS ribozyme (Klein and Ferré-D'Amaré, 2006) and VS ribozyme (Lilley, 2004) require interaction with specific proteins to become active. Another example of RNA forming quaternary structure with proteins is the *ribosome*, which consists of multiple rRNAs, supported by a family of proteins called the rProteins (Nissen et al., 2001). In (Miao and Westhof, 2015) a probabilistic method of characterizing the complex nature of the interactions between RNAs and proteins is described.

## 2.3 RNA secondary structure prediction

RNAs are chemical species and their folding into secondary and tertiary structures is governed by the fundamental laws of physics and thermodynamic principles. Given a single or a set of evolutionarily related RNA sequences, the problem of secondary structure prediction is to identify the minimum free energy structure of the sequence(s). However, RNAs are not static molecules trapped in a single structure and they often transition from one stable structure to a slightly different stable structure due to fluctuations of environmental conditions such as the temperature. Thus probabilistic frameworks based on Boltzmann partition function computation methods have also been developed to characterize the ensemble of all possible secondary structures of a given RNA sequence. Over the past 35 years, algorithms for secondary structure prediction using experimentally measured free energy parameters, often referred to as the nearest neighbour energy models, have been developed. In this section we give a comprehensive overview of the foundations of RNA secondary structure modeling and the existing algorithmic approaches for single sequence and multiple sequence RNA secondary structure prediction.

### 2.3.1 Nearest-neighbour energy models

The free energy of an RNA secondary structure at 37°C, $\Delta G^{\circ}_{37}$ can be computed using the empirical nearest neighbour parameters (Dirks and Pierce, 2003; Mathews et al., 1999). In a nearest neighbour model the thermodynamic stability of each type of SSE depends on the identity of its neighbouring nucleotides. For a given RNA at equilibrium, there is an equilibrium between strands folded in structure $\tau_1$, and the unstructured random coil (RC) state

$$RC \rightleftharpoons \tau_1 \tag{1}$$

where the equation is governed by the equilibrium constant $K_1$:

$$K_1 = \frac{[\tau_1]}{[RC]} \tag{2}$$

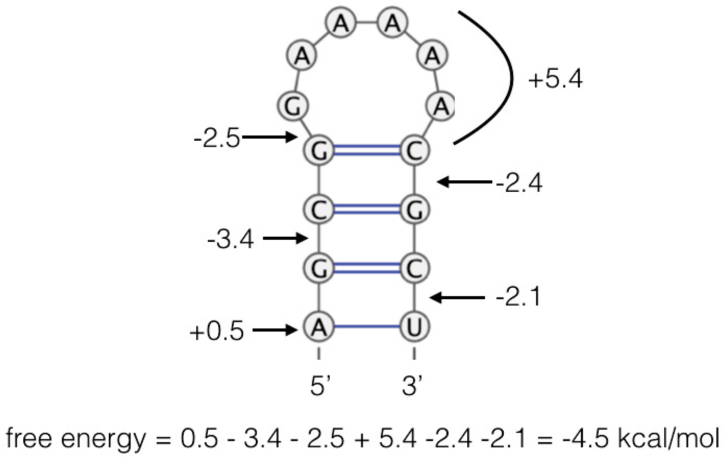
For structure $\tau_1$ the free energy change $\Delta G_{37^\circ}(\frac{[\tau_1]}{[RC]})$ quantifies the stability of the structure by the relationship

$$\frac{K_1}{K_2} = \frac{\tau_1}{\tau_2} = e^{(\Delta G_{37^\circ}(\frac{[\tau_1]}{[RC]})-\Delta G_{37^\circ}(\frac{[\tau_2]}{[RC]})/RT} \tag{3}$$

It follows that the lowest free energy structure is the most represented conformation at equilibrium. Figure 7 gives an example of how for a given RNA sequence $\phi$ and secondary structure $\tau$ the free energy can be computed using the energy parameters specified by the Turner energy model (Mathews et al., 1999). The Turner energy model does not include the energy parameters required to describe any form of pseudoknots. However, some other energy models such as (Bindewald et al., 2011; Dirks and Pierce, 2003) do provide the required parameters to describe the energy contribution of a subclass of all possible pseudoknots.

### 2.3.2 Sequence and structure space

For an RNA sequence $\phi$ of length $N$, there are $4^N$ possible combinations of nucleotides. It has been shown that for a given sequence the total number of structures with minimum free energy is roughly $1.8^N$ (Zuker and Sankoff, 1984). Notably for a given sequence, the MFE structures are not uniformly distributed; that is, some MFE structures are more abundant than others while



A
A
A
A
A
G
G
C
C
G
G
C
A
U
+5.4
-2.5
-2.4
-3.4
-2.1
+0.5
5'
3'
free energy = 0.5 - 3.4 - 2.5 + 5.4 -2.4 -2.1 = -4.5 kcal/mol

Figure 7: A nearest-neighbour calculation of $\Delta G_{37^\circ}$ for a stem-loop structure. Each free energy increment is shown. The negative values indicate stabilizing effects and positive values indicate destabilizing effects. The total stability, -4.5 kcal/mol, is the sum of the increments.

some are quiet scarce (Aguirre et al., 2011). For a sequence of length 100, the total number of possible secondary structures is $3.4 \times 10^{25}$. If a single computer processor can compute the free energy for $1 \times 10^4$ structures per second, calculating the free energy for each possible secondary structure explicitly, would require $1.1 \times 10^{14}$ years. Consequently a brute force approach is not a viable option to find the MFE structure.

### 2.3.3 Free energy minimization

The nearest-neighbour free energy calculation method provides the means to compute the free energy $\Delta G(\phi, \tau)$ of RNA sequence $\phi$ folding into secondary structure $\tau$. Using computational methods, one can find the MFE structure $\tau_{MFE}$ of a sequence $\phi$. One approach to find MFE for $\phi$ is to explicitly generate the set of all possible secondary structures and corresponding free energies associated with $\phi$. As discussed in section 2.3.2, for a sequence of length 100 there are approximately $3.4 \times 10^{25}$ different secondary structures and a brute force approach to find the MFE structure is not feasible.

The first algorithm to find the MFE structure was published by Nussinov and Jacobson (Nussinov and Jacobson, 1980) follows a dynamic programming approach. Dynamic programming algorithms implicitly check all possible secondary structures without explicitly generating each individual structure. A dynamic programming algorithm divides the problem into a large number of smaller problems and uses recursion to build the solution to the complete problem. Two steps are used to predict the lowest free energy structure. In the first step, called the fill step (the slower of the two steps), the lowest free energy of secondary structure formation is calculated and stored for each sub-fragment of the total sequence, starting from short fragments and then progressively calculating the lowest free energy of folding for longer fragments by joining the smaller fragments.

At the end of the fill step, the lowest free energy for a structure from the given sequence is known, but the structure itself is yet unknown. The second step of the calculation is called the traceback step and determines MFE by backtracking through the free energies of the subsequences. Dynamic programming methods guarantee that the MFE structure is found given the secondary structure topologies that are realized by the nearest-neighbour energy models used. A popular program named `RNAfold` which is part of the RNA Vienna package (Hofacker, 2003) implements the dynamic programming method algorithm introduced by (Zuker and Stiegler, 1981) and finds the MFE structure of a given RNA sequences in $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ space. A review of dynamic programming techniques to find the MFE structures can be found elsewhere (Eddy, 2004).

The Nussinov algorithm and `RNAfold` find the MFE in $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ space, however they can not predict pseudoknots. Since the general problem of predicting pseudoknotted secondary structures is NP-hard (Akutsu, 2000), several algorithms have been proposed that find the MFE secondary structure from a restricted class of secondary structures (Condon et al., 2004). Eddy et al. (Rivas and Eddy, 1999) devised a dynamic programming algorithm in $\mathcal{O}(n^6)$ time and $\mathcal{O}(n^4)$ space to cover a wide class of pseudoknots. To improve on the prohibitive run-time requirements of Eddy et al., subsequent dynamic programming methods to predict the MFE structure such as `NUPAK`, `RNAStructure` and `pKiss` were developed to find the pseudoknotted MFE structure in $\mathcal{O}(n^5)$, $\mathcal{O}(n^4)$ and $\mathcal{O}(n^4)$ time respectively (Akutsu, 2000; Dirks and Pierce, 2003; Theis et al., 2010). However the problem of finding the MFE structure including all possible types of pseudoknots has been shown to be NP-complete (Lyngsø and Pedersen, 2000) and therefore `NUPAK`, `RNAStructure` and `pKiss` each can only cover a small class of pseudoknotted structures.

Other than dynamic programming approaches, other classes of algorithms have also been utilized to find the MFE structures. Chen et al. (Chen et al., 2009) implement a heuristic to find the MFE structure including a wide class of pseudoknots in $\mathcal{O}(n^5)$ time. Another method named `IPknot` implements an integer programming technique to assemble small pseudoknot-free structural as predicted by `NUPACK` and then rapidly assembles them to find the unified MFE structure including pseudoknots (Sato et al., 2011).

### 2.3.4 Suboptimal solutions

The accuracy of RNA secondary structure prediction by free energy minimization is limited by several factors. First, the free energy nearest-neighbour models are incomplete as some sequence effects on stability can not be fully described by the nearest-neighbour model (Longfellow et al., 1990). Second, many functional RNAs such as ribozymes and riboswitches can transit from one conformation (i.e., inactive conformation) to another conformation (i.e., active conformation) in response to environmental stimuli (Schultes and Bartel, 2000; Serganov and Nudler, 2013). Third, RNAs at equilibrium do not always fold into the MFE structure as folding kinetics may affect the folding process (Treiber and Williamson, 2001). These limitations justify the need for methods that can predict low energy sub-optimal secondary structures. Zuker et al (Zuker, 2003) implemented `mfold` to generate a diverse set of sub-optimal secondary structures for a single RNA sequence.

### 2.3.5 Base pair partition functions

Another rigorous approach to characterize an ensemble of all possible secondary structures associated with RNA sequence $\phi$ is through computation of the partition function. In physics, a partition function describes the statistical properties of a system in thermodynamic equilibrium. In 1990 McCaskill (McCaskill, 1990) derived a set of recursions and used dynamic programming techniques to compute the partition function over the ensemble of all possible secondary structures without pseudoknots:

$$Q\left(\phi\right) = \sum_{\tau \in \Gamma} e^{-\Delta G(\phi,\tau)/k_B T} \tag{4}$$

where $\Gamma$ is the ensemble of all possible secondary structures, $k_B$ is the Boltzmann constant and $T$ is the absolute temperature. In the fill stage of the dynamic programming method, partition functions are determined for all sequence fragments, starting with the shortest. The backtrack step is similar to the backtracking step of the Zuker method.

Given the partition function, one can compute the the probability of sequence $\phi$ folding into structure $\tau$ at equilibrium by

$$p\left(\phi,\tau\right) = \frac{1}{Q\left(\phi\right)} e^{-\Delta G(\phi,\tau)/k_B T} \tag{5}$$

The partition function provides the means to compute the statistical properties of an RNA sequence and the corresponding ensemble of all possible secondary structures. However, the partition function does not predict the MFE or sub-optional secondary structures. McCaskill's algorithm computes the partition function in $\mathcal{O}(n^3)$ time for pseudoknot-free ensembles. The `NUPACK` package includes a set of recursions that can realize a sub class of possible pseudoknots and can compute the partition function. To our knowledge `NUPACK`, is the only existing method to compute the partition function for pseudoknotted structures. `NUPACK` runs in $\mathcal{O}(n^5)$ time and $\mathcal{O}(n^4)$ and was developed in 2003 by Dirks and Pierce et al. (Dirks and Pierce, 2003). Figure 8 following figure presents the class of pseudoknots the Dirks and Pierce model can realize. Figure 9 shows the pseudoknots not supported by Dirks and Pierce model.

### 2.3.6 Statistical sampling

Ding and Lawrence introduced a new approach based on statistical sampling techniques they first developed in 1999 (Ding and Lawrence, 1999). Based on the idea of statistical sampling Ding and Lawrence devised a dynamic programming method to efficiently sample sub-optimal secondary structures from the Boltzmann ensemble of all possible pseudoknot-free secondary structures of a given RNA sequence (Ding and Lawrence, 2003). The fill step of the algorithm is similar to McCaskill's algorithm however the backtracking step is different. In the backtracking step the base

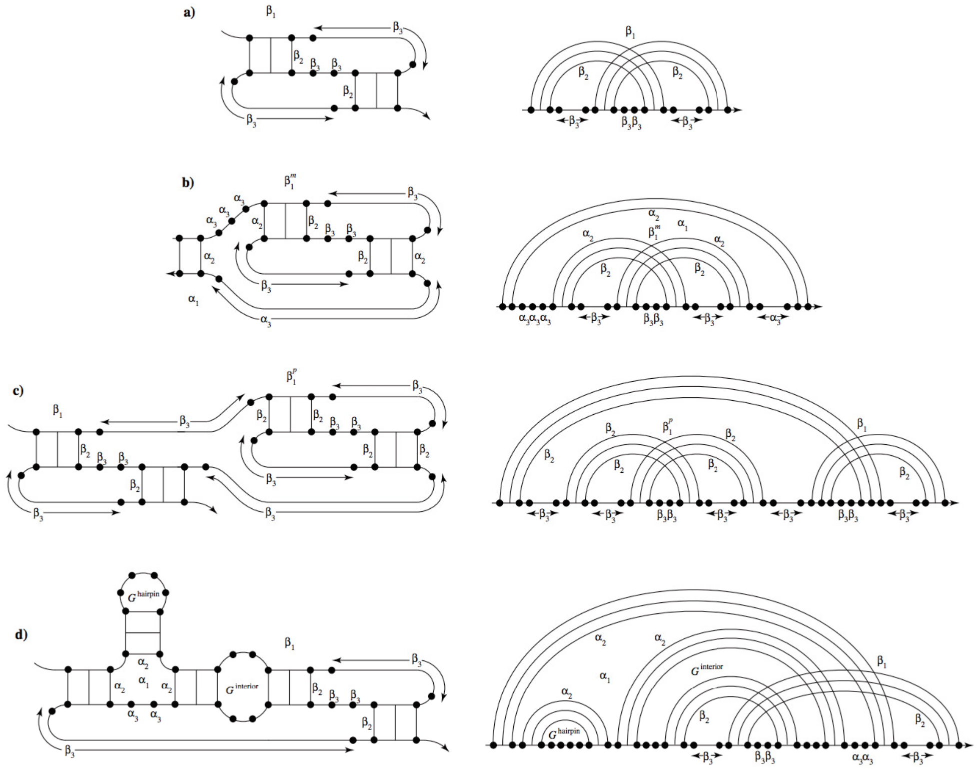


Figure 8: a) external pseudoknots b) pseudoknot inside loop c)pseudoknot inside pseudoknot d) pseudoknot with a hairpin and an interior loop inside a spanning region of the pseudoknot. $\alpha_i$ and $\beta_i$ quantify the energy contribution of formation of pseudoknots at each region (Dirks and Pierce, 2003).

pairs are chosen probabilistically based on the partition function of all possible sequence fragments. Overall, the probability of sampling any structure is equal to the probability of its occurring in the Boltzmann ensemble as characterized by equation 5. The statistical sampling algorithm of Ding and Lawrence is implemented in the `Sfold` software package (Ding et al., 2004).

### 2.3.7 Secondary structures common to multiple sequences

When multiple homologous RNA sequences are available, one can find the consensus secondary structure that is common to all the sequences. First, multiple sequence alignment (MSA) methods are used to find the nucleotides that are conserved and are common to the sequences and then the consensus secondary structure is derived.

There are two main approaches to find the secondary structure common to a homologous set of sequences. The first approach is to predict the structure common to multiple sequences in a fixed alignment. `Alifold` (Hofacker et al., 2002) is a dynamic programming algorithm for predicting the lowest free energy pseudoknot-free structure common to a sequence alignment. `Alifold` runs in $\mathcal{O}(A^3)$ where $A$ is the number of aligned sequences. `Alifold` can be also used to compute the

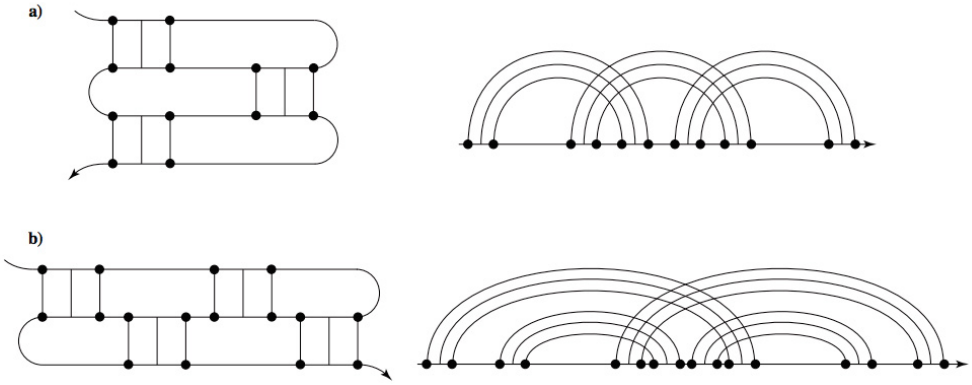


Figure 9: neither of the above are supported by Dirks and Pierce energy model (Dirks and Pierce, 2003).

partition function. Another algorithm named ConStruct (Hofacker et al., 1994) predicts the base pair probabilities for each sequence of the alignment separately. Then ConStruct uses the sequence alignment to find consensus base pair probability by computing the sum of the probabilities of the sequences. ConStruct runs in $\mathcal{O}(N_1^3+...+N_m^3)$ where $N_i$ is the length of the $i^{th}$ sequences. Another method named hxmatch computes the consensus structures including pseudoknots based on alignments of a few sequences. The algorithm combines thermodynamic and covariation information to assign scores to all possible base pairs, the base pairs are chosen with the help of the maximum weighted matching algorithm (Witwer et al., 2004). The run-time requirement of hxmatch is $\mathcal{O}(L^3)$ where $L$ is the length of the MSA. To our knowledge, hxmatch is the only method that can be used to predict a consensus secondary structure common to multiple sequences and can include a sub class of pseudoknots.

The second approach to find the consensus secondary structure common to multiple sequences is to simultaneously find the optimal secondary structure and alignment. Sankoff developed a dynamic programming algorithm to simultaneously determine the lowest free energy structure common to multiple sequences and the sequence alignment that facilitates the common structure (Sankoff, 1985). The Sankoff method runs in $\mathcal{O}(N_1^3 \times ... \times N_m^3)$ and does not scale well. FOLDALIGN implements a dynamic programming algorithm to find locally conserved base pairing motifs of up to L nucleotides using a scoring function based on nucleotide identities and runs in $\mathcal{O}(L^4)$ (Gorodkin et al., 1997). A major limitation of FOLDALIGN is that it excludes k-way loops and can only consider two sequences in the alignment. Neither the Sankoff method nor FOLDALIGN can include pseudoknots in the prediction.

## 2.4 RNA tertiary structure prediction

The next step after predicting secondary structure is the tertiary structure prediction. Compared to the methods to predict protein tertiary folding, RNA tertiary or 3D folding is still in its infancy and existing methods are limited to the size as well as the type of the topological configurations. RNA 3D folding methods often use the secondary structure as well as the 3D contacts information to guide the 3D folding process. In this section we give an overview of the most popular computational approaches for RNA 3D structure prediction.

The nucleic acid simulation tool or `NAST` is a molecular dynamic simulation tool (Jonikas et al., 2009) using a coarse-grained model with resolution of one bead per nucleotide. `NAST` requires secondary structure information and, if available, accepts tertiary contacts to direct the folding. When only the secondary structure is used, the quality of prediction remains limited to simple structures such as hairpin loops. When tertiary contacts are included, `NAST` can predict the 3D structure of large RNAs of up to 160 nucleotides with average root-mean-square deviation (rmsd) of 8Å.

The `iFoldRNA` web server (Krokhotin et al., 2015) offers a platform that combines experimental data, secondary structure information and molecular dynamics simulations to predict tertiary structures of RNA as long as a few hundred nucleotides with atomic level detail. It takes about one day of computation for `iFoldRNA` to predict the 3D structure of the M-box riboswitch of length 161 nucleotides with 7.7Å rmsd between the predicted and the crystal structure.

`FARFAR` (Das et al., 2010) utilizes a fragment assembly phase followed by a refinement phase of atomic level interactions. `FARFAR` is limited with prediction of 3D structures of small RNA molecules of size 6-20 nucleotides. `FARFAR` shows excellent accuracy of less than 1Å rmsd with the crystal structure for simple shapes, however for more complex structures such as k-way junctions the quality of prediction drops by approximately 10 fold.

`MC-Fold` and `MC-Sym` pipeline (Parisien and Major, 2008) introduces the notion of nucleotide cyclic motif (NCM) as a new way to represent nucleotide relationships in structured RNAs. The `MC-Fold` and `MC-Sym` pipeline uses integer programming to assemble NCM fragments to predict RNA 3D structures. The `MC-Fold` and `MC-Sym` pipeline has showed promising performance of 1.7 to 2.9 Å rmsd with the crystal structure for RNAs of length 18 to 47 nucleotides.

`RNA-MoIP` (Reinharz et al., 2012) takes the RNA 3D prediction to the next step by improving on quality and size of the RNA molecules as well as the run-time requirements. `RNA-MoIP` replaces the `MC-Fold` portion of the `MC-Fold` and `MC-Sym` pipeline. `RNA-MoIP` uses an integer programming framework to assemble secondary structure motifs into sub-optimal RNA secondary structures. In the next step `RNA-MoIP` uses the generated suboptimal structures and the related secondary motifs as input to `MC-Sym`. `RNA-MoIP` has shown significant reduction in rmsd values and run-time requirements compared to the other methods and have produced the best of results for RNAs of up to 150 nucleotides.

The use of secondary structure information as a preprocessing step in all of the methods listed here, highlights the importance of secondary structure prediction in understanding RNA function as well as in designing novel RNA structures with desired function.

## 2.5 RNA secondary structure design

### 2.5.1 Problem Statement

RNA secondary structure design or the RNA inverse folding problem is the process of finding RNA sequences that are predicted to fold into a desired secondary structure as predicted by a folding algorithm. More precisely, the classical definition of RNA inverse folding is to find the sequence $\phi$ that is predicted to have a targeted secondary structure $\tau$ as its MFE structure, that is $MFE(\phi) = \tau$.

For targeted secondary structure $\tau$ of length $n$, the total number of possible secondary structures is $4^n$ and therefore a brute-force approach to find sequence $\phi$ such that $MFE(\phi) = \tau$ is not feasible. By observing that each position on $\phi$ is either paired or unpaired and that only canonical base pairs are allowed to be formed, the total number of sequences that are compatible with a secondary structure $\tau$ is given by

$$(6^{p/2})(4^u) \tag{6}$$

where $p$ is the number of paired positions and $u$ is the number of unpaired positions. For instance a hammerhead ribozyme of length 81 nucleotides with 20 paired and 41 unpaired positions is compatible with $1.768 \times 10^{40}$ sequences.

## 2.6 RNA secondary structure design as a combinatorial optimization problem

The problem of finding a set of RNA sequences that conforms the target secondary structure, falls in the category of combinatorial optimization problems (COPs). Almost all of the existing computational methods to solve RNA design, model the problem as a COP and use related computational techniques to develop the design approach. Before diving into RNA design, in the next chapter we describe the formal definition of COPs and describe RNA design as a COP. Then we review some of the most well-known instances of COPs and give an overview of the existing methods to solve COPs.

# Chapter 3

# Combinatorial Optimization Problems

## 3.1 Preface

Numerous applications in computer science, biology, chemistry, physics and engineering can be described and solved in some form of a combinatorial optimization problem (COP). Indeed the RNA secondary structure design problem is classically described as a very particular instance of COPs. Many practical COPs are complex and finding solutions for them is not trivial and general mathematical solutions are not available and the vast search space of the problem can not be effectively visited with brute force methods.

In this chapter we describe a formal definition of COPs and describe several well-known examples of COPs including the RNA inverse folding problem. Then we provide a comprehensive overview of the different classes of algorithms to solving COPs. We describe several classes of heuristic methods, stochastic, evolutionary and memetic approaches.

## 3.2 Definition

A combinatorial optimization problem $P$ can be described as either a minimization problem or a maximization problem, and can be described by the following elements:

1. finite set $D_P$ of instances
2. finite set $S_P(I)$ of candidate solutions for each instance $I \in D_P$
3. function $f$ which assigns a positive number $f(I,s) \in \mathbb{R}$ called the solution value for $s$ to each instance $I \in D_P$ and each candidate solution $s \in S_P(I)$

$s^\star \in S_P(I)$ is an optimal solution for a problem instance $I$ such that $\forall s \in S_P(I)$, $f(I,x^\star) \leq f(I,x)$ if $P$ is a minimization problem, and $f(I,s^\star) \geq f(I,s)$ if $P$ is a maximization problem.

Note that the problem $P$ has a finite number of possible instances and therefore at least one solution $s^\star$ exists which —in theory —can be found using a brute force approach. However for many interesting derivations of COPs, the size of the search space grows exponentially as the size of the problem grows and therefore an exhaustive enumeration of all possible solutions becomes impractical. For instance, as discussed in section 2.5.1, the search space to design a small RNA molecule of length 81, there are $1.768 * 10^{40}$ possible solutions. For a large class of COPs, no polynomial algorithms to find the optimal solution is known. The computational complexity theory (Hopcroft et al., 2006) and in particular the theory of NP-completeness (Garey et al., 1976) provide the means to characterize such problems and the literature is plentiful of specialized algorithms to find near optimal solutions for COPs with exponentially growing search space of arbitrary size.

## 3.3 NP completeness theory

The NP-completeness theory tries to answer the decision problem of whether for minimization problem $P$ it is possible to find solution $s \in S_p(I)$ such that $f(I, s) \leq T$ where $T$ is an arbitrary threshold.

One can identify two distinct classes of the above decision problem. The first class called the $P$ class is the class of decision problems which can be solved by polynomial time algorithms. The second class called the NP class that can be solved by non-deterministic polynomial time algorithms. Solving NP problems consists of two stages. First a solution is guessed and then the solution is checked using a deterministic polynomial time algorithm.

Given the two basic classes of decision problems discussed above, the class of NP-complete problems (Hochbaum, 1982; Lin, 1965) can be defined as following:

**Definition 3.1**: the decision problem $P$ is NP-complete id a) $P \in NP$ and b) all problems in NP can be reduced to $P$ using a polynomial mapping function $M$.

From the above definition is follows that if for one problem in NP a polynomial algorithm to solve can be found, then all problems in NP can also be solved in polynomial time. Perhaps one of the most important questions in computer science is to show whether P = NP or not. Despite the common belief is that P $\neq$ NP (Baker et al., 1975; Fortnow, 2009), there exists no proof to neither accept nor to reject this.

COPs and generally speaking optimization problems are not decision problems and therefore can not be NP-complete. Optimization problems belong to a less restrictive class of problems named NP-hard problems (Cheeseman et al., 1991).

**Definition 3.2**: problem is NP-hard if all problems in NP are polynomially reducible to it.

It follows that problems that are not necessarily NP can be NP-hard. Due to the high complexity of the search space of NP-hard problems, tailored approximation algorithms (Hochbaum, 1996) are developed that are not guaranteed to find the optimal solution but are quiet powerful and can find near optimal solutions in reasonably short amounts of time. The RNA design problem is known to be NP-hard (Schnall-Levin et al., 2008).

## 3.4 Examples of COPs

There is a wealth of different COPs in the literature (Aissi et al., 2009; Bianchi et al., 2009; Graham et al., 1979; Oliveira and Pardalos, 2005; Smith, 1999). In this section we touch upon some of the most well-known ones.

### 3.4.1 The traveling salesman problem

One of the very well known instances of COPs is the traveling salesman problem (TSP) (Flood, 1956). Given a set of $n$ cities and the Euclidean distances between them, the TSM has to find the shortest path in which he visits all the cities exactly once and finally ends up at his starting point. The goal is to minimize

$$l(\pi) = \sum_{i=1}^{n-1} d_{\pi(i),\pi(i+1)} + d_{\pi(n),\pi(1)} \tag{7}$$

where $l(\pi)$ represents the length of the traveled path, $d_{i,j}$ is the Euclidean distance between cities $i$ and $j$ and finally $\pi$ is a permutation of $\langle 1, ..., n \rangle$. An instance $I$ of the TSP can be defined by a square matrix D and a solution can be represented by permutation $\pi = \left\{..., \pi_j^i, ...\right\}$ where $\pi_j^i$ represents city $j$ at the $i$th step. Despite being easy to describe, the TSP is NP-hard (Woeginger, 2003).

The TSP has applications in X-ray crystallography (Bland and Shallcross, 1989), clustering of data arrays (Lenstra and Kan, 1975), prediction of protein function (Johnson and Liu, 2006) and robotic path planning (Yu et al., 2002) to name a few.

### 3.4.2 The knapsack problem

The knapsack problem (KP) is another well known COP. Given a knapsack which can be used to transport a number of items with a maximum total weight, the task is to select a subset of all items such that the value of the items is maximized while respecting the weight limit. The KP can be formalized as follows:

$$\text{minimize} \quad \sum_{i \in k} c_i \tag{8}$$

$$\text{subject to} \quad \sum_{i \in k}^{n} w_i \leq W, \quad K \subset \{1, ..., n\} \tag{9}$$

where $c_i$ denotes the value of item $i$, $w_i$ the weight of item $i$, and $W$ the weight capacity of the knapsack. An instance of the KP is defined by the tuple $I = \langle c, w, W \rangle$ where $c = (c_1, ..., c_n)$, $w = (w_1, ..., w_n)$ and $W \in \mathbb{R}$

The general version of the KP problem called the generalized knapsack problem can is defined by

$$\text{minimize} \quad \sum_{i \in k} c_i \tag{10}$$

$$\text{subject to} \quad \sum_{i \in k}^{n} w_{ij} \leq W_j, \quad K \subset \{1, ..., n\}, \quad \forall j = 1, ..., m \tag{11}$$

Here the weight of an item as well as the maximum weight capacity of the knapsack have $m$ dimensions.

The knapsack problem has applications in cognitive radio networks (Song et al., 2008), scheduling (Babaioff et al., 2007; Kellerer and Strusevich, 2010), and container shipping network design (Shintani et al., 2007) to name a few.

### 3.4.3 The RNA secondary structure design problem

RNA design as a COPs has been shown to be NP hard (Bonnet et al., 2017). The classical definition of the RNA design problem can be formalized by:

$$\text{minimize} \quad \Delta(MFE(\phi) - \tau) \tag{12}$$

where $\tau$ is the target secondary structure which we would like to design, $\phi$ is the designed sequence which is predicted to have $\tau$ as its minimum free energy (MFE) structure and $\Delta$ represents the tree edit distance between $\phi$ and $\tau$. A solution for this problem can be represented by permutation $\phi = \langle \phi_i, ..., \phi_n \rangle$ where $|\phi| = |\tau| = n$.

RNA molecules have size of nano scale. Hence characterizing the state of an RNA molecule requires using probability distributions. With that in mind, one approach to modelling the RNA design problem is to use a probabilistic approach to describe the objective function. For instance, optimizing for Boltzmann probability is a popular approach. Using probabilistic objective functions puts the RNA design into the class of stochastic COPs (SCOPs). We will give formal description for SCOPs later in this chapter.

The RNA secondary structure design problem is particularly interesting. In many well-known COPs computing the cost function is a linear operation. However, in the RNA design problem, depending on whether pseudoknots are considered or not and also depending on the optimization criteria (MFE distance optimization versus Boltzmann probability optimization), the cost function may have computational and storage complexity of $\mathcal{O}(n^3)$ to $\mathcal{O}(n^6)$. The high computational cost associated with computing the cost function makes exploration of even a small neighbourhood of the search space of a large problem instance (i.e $n > 100$) infeasible. The exponential growth in size of the search space as well as the exponential growth in computational complexity of computing the cost function makes the RNA design problem a particularly challenging and interesting one. Variants of RNA secondary structure design problems where different formulations than the one presented by equation 12 are extensively presented in Chapter 4. The RNA design problem has applications in nano technology (Petros and DeSimone, 2010), synthetic biology (Isaacs et al., 2006), therapeutics (Ding, 2010; Edelstein et al., 2007; Leonard and Schaffer, 2005) and materials design (Afonin et al., 2010).

## 3.5 Exact methods for solving COPs

Despite the brute and force search strategy being impractical to solve optimization problems where the search space grows exponentially, there are numerous algorithmic methods that can guarantee finding optimal or near optimal solutions in reasonable amounts of time.

### 3.5.1 Branch and bound

The branch and bound method (Lawler and Wood, 1966) for finding optimal solution for COPs consists of finding lower and upper bounds for the optimal solution, as well as a schema to navigate through the search space efficiently. Assuming a minimization problem, the upper bounds are often found by heuristics that find near optimal solutions in short amounts of time. To find the lower bounds, the problem is relaxed by removing at least one of the constraints. The navigation of the search space splits the problem into child subproblems in such a way that the union of the solutions to the child problems generates solutions for the parent problem. Each subproblem will be recursively divided into more subproblems generating a branching tree. The recursive division of a subproblem stops when a solution to the the subproblem is found. A solution of a subproblem is found when the lower bound is equal to the upper bound or when the lower bound is above the best feasible solution found so far. The branching tree can potentially grow in size exponentially. To avoid exponential growth of the branching tree, one is required to find effective heuristics for finding upper bounds and also to use effective relaxation techniques to produce lower bounds. In COPs, discrete relaxation techniques are used to generate lower bounds (Fisher, 1981).

The branch and bound method has been successfully used in solving some challenging instances of COPs (Mezmaz et al., 2007; Mitchell, 2002; Padberg and Rinaldi, 1987; Visée et al., 1998). Notably, in many COPs finding a suitable discrete relaxation procedure is challenging and therefore finding good enough lower bound is also a significant challenge. In circumstances where good lower bounds can not be found, the best the branch and bound method can do is to find approximate solutions within a particular range from the optimal solutions.

### 3.5.2 Branch and cut

Another approach called branch and cut (Padberg and Rinaldi, 1991) is based on the idea of finding a relaxation in form of a linear program which has the same optimal solution as the original problem. The branch and cut method is an exact algorithm which is guaranteed to find the optimal solution. The linear programming (LP) problem (Dantzig, 2016; Luenberger, 1973) can be formalized by:

$$\text{minimize} \quad C^T x \tag{13}$$

$$\text{subject to} \quad Ax \leq b, \quad x \leq 0 \in \mathbb{R} \tag{14}$$

where $x$ and $c$ are n-vectors and $b$ is m-vector. It follows that $A$ is $m \times n$ matrix. Given the above linear system, the simplex algorithm (Winston and Goldberg, 2004) is guaranteed to find the optimal solution. The simplex algorithm systematically searches the extremities of the polyhedron $P$ defined by the inequalities given by the LP.

To solve a COP with linear programming techniques, the search space is enlarged by extending the solution often from binary vectors to vectors of continuous variables. Since not all facets of the polyhedron $P_C$ are known for every combinatorial problem or the number of facets is simply too high, a cutting plane approach has been developed. This approach works as follows

- an initial polyhedron $P_C \subseteq P$ is generated so that the LP can be solved in reasonable time
- then an LP solver is used to generate a solution $x^*$
  - if $x^*$ represents a feasible solution to the COP, it implies that the optimum solution is found and then the algorithm terminates
  - if $x^*$ does not represent a feasible solution to the COP then
    - the algorithm searches for a cut such that $x^*$ is cut off the polyhedron by ensuring that the new polyhedron still contains the polyhedron of the COP
    - the inequality found is added to the system of equations and the resulting LP is solved to obtain a new $x^*$

These steps are repeated until the optimal solution is found or the algorithm fails to find a new feasible cut. Since the latter case is more likely to occur, a branching rule can be used to split the problem into subproblems and the cutting plane procedure can be applied recursively to the subproblems. This entire process is called the branch and cut method.

The branch and cut algorithm has been successfully applied to solve COPs such as large-scale symmetric traveling salesman problem (Padberg and Rinaldi, 1991), the capacitated vehicle routing problem (Lysgaard et al., 2004), vendor-managed inventory routing problem (Archetti et al., 2007), location rounting (Prodhon and Prins, 2014), protein structure alignment (Lancia et al., 2001) and graph coloring (Méndez-Díaz and Zabala, 2006) to name a few.

### 3.5.3 Heuristic methods

Heuristics are search methods that find near optimum solutions to optimization problems in short times. In comparison to exact approaches, heuristics are not guaranteed of finding optimum solutions nor do they generally provide a guarantee to find solutions within a certain range to the optimum. Nevertheless, heuristics are powerful methods to generate high quality solutions for diverse ranges of COPs of practical size and interest (Burkard and Rendl, 1984; Dorigo and Di Caro, 1999; Geem et al., 2001; Pearl, 1984). Many heuristics have the advantage of being applicable to a wide range of problems so it is often relatively easy to devise heuristics to find quality solutions for COPs. The developed heuristics can often be easily modified to adapt for changes in the objective function. Heuristic methods provide the flexibility to later on easily add extra constraints to the problem solver. Generally speaking, the COPs are complex and often exact methods such as branch and bound or branch and cut are not applicable. Moreover even in cases where exact methods such as branch and bounds are applicable, one is still required to devise proper heuristics to find the upper bounds.

## 3.6 Nature inspired methods

Many algorithms for solving COPs are nature inspired, and have been developed by drawing inspiration from nature. By far the majority of nature inspired algorithms are based on some successful characteristics of biological systems. In this section we give a brief overview of some of the most well known biologically inspired methods for solving COPs.

### 3.6.1 Evolutionary algorithms

Inspired by the idea of natural evolution several evolutionary algorithms (EA) have been proposed and successfully used for solving diverse ranges of optimization problems. The source of inspiration for EAs goes back to Darwin's theory about existence and evolution of life on earth (Darwin and Bynum, 2009). According to Darwin, the three fundamental concepts playing key roles in evolution are replication (recombination), variation (mutation) and natural selection (survival of the fittest). The idea is that due to a scarcity of resources, individuals within a species must compete and therefore the fitter ones have more likelihood to gain access to the resources which leads to higher chances of survival.

From an information processing point of view, evolution can be regarded as an optimization process where the species (i.e problem instances) evolve to improve on their fitness and therefore improve on their chances for survival (i.e., finding solutions). Within each species each organism carries its genetic information which is referred to as the genotype. The organism's traits constitute the phenotype. The genetic information is eventually passed on to the next generation if the organism reproduces before it dies. While replication combined with variation allows for improving the genetic information, natural selection implicitly evaluates the fitness of each phenotype which leads to improvement on its odds for survival. Based on the ideas taken from recombination, mutation and survival of the fit, different flavours of EAs have been developed.

**Evolutionary Strategies (ESs)** are a family of EAs first introduced in 1960 (Schwefel, 1975) for continuous real valued parameter optimization. The idea was to perform mutation and selection on a two-membered population in iterative steps. Later on the ESs themselves evolved to allow multi-parent recombination as well as using alternative selection strategies.

**Evolutionary Programming (EPs)** are a family of EAs first introduced in 1966 (Fogel et al., 1966) based on mutation and selection on finite state machines.

**Genetic Algorithms (GAs)** are a family of EAs first invented by Holland (Holland, 1992) where the initial idea was to study the phenomenon of adaptation in nature with the aim of defining a framework to use the natural adaptation in computer systems to solve optimization problems. In the Holland's model genomes were described as strings of zeros and ones. The nature-inspired operators, namely crossover and mutation, were being applied on the chromosomes and then a selection operator was applied to filter the population of offspring. The crossover operation was first used by Hollands GA.

**Outline of EAs** The different flavours of EAs all follow similar algorithmic approaches and only differ in some of the technical details. Figure 10 illustrates the overall flow of a generic EA.

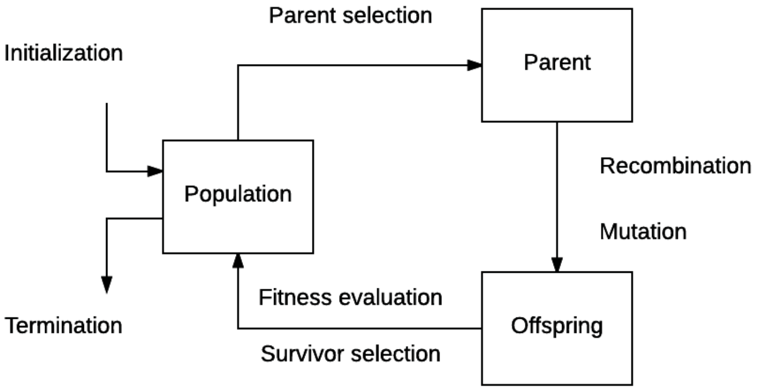


Figure 10: General schema of Evolutionary Algorithms

Before designing an EA to solve a problem, one must first define a representation for the individual objects that are going to be subject to evolution. Object-forming possible solutions within the original problem context are referred to as phenotypes and their encodings within the context of the EA are called genotypes. A representation of an object is a mapping from its phenotype onto a set of genotypes that are said to represent the phenotypes. For instance, given an optimization problem on integers, then given set of integers represent the phenotypes. To model the phenotypes as input for the EA, one could map the integers into genotypes by transforming each into into its binary representation. For instance integer 2 can be represented by 01. Note the phenotype space can be very different from the genotype space and that the evolutionary process happens inside the genotype space. Once the evolution comes to an end, a good solution —a phenotype —is obtained by decoding the corresponding genotype back into the phenotype space.

Once we define the genotype space for the problem, the next task is to generate a population of genotypes. The initial population is often generated randomly, however, often promotes domain knowledge into the genotype space of the initial population, some heuristics are used during the initialization stage.

Given a population of genotypes, the next step is to choose parents. The role of parent selection is to distinguish among individuals based on their quality to allow better individuals to become parents of the the next generation. An individual genotype is called a parent if it is selected for recombination with one or more other parents. The first time the parent selection happens is often random since there is no notion of fitness or quality attached to the individuals. However, after one generation the parent selection becomes probabilistic to give fitter individuals higher chances for recombination. In a typical EA two parents combine in a process called recombination to generate a third genotype referred to as the offspring. Each offspring will have a probability of observing one or more random mutations in its genotype representation. A commonly used mutation operation called the unary mutation —when applied to an offspring —delivers a slightly changed mutant.

Once a population of offspring is generated, the quality of each genotype is measured in a process called fitness evaluation. The role of fitness evaluation is to characterize the requirements to which the EA needs to adapt. Fitness evaluation provides the basis for selection of fit individuals and facilitates overall improvements in the quality of the population. Fitness evaluation is carried by a function called the fitness function and its role is to compute a quality measure for each individual. For instance if we are maximizing integer $x$, then the fitness of individual 01 will be 2. The fitness of each individual genotype will be attached to it once computed by the fitness function. At this step if the desired fitness value has emerged, the evolutionary cycle will stop and the genotype with desired fitness will be decoded to the corresponding phenotype and then the phenotype is returned.

The last step of each evolutionary cycle is to choose a set of survivals. The survivals are selected from the pool of offsprings and parents all together. In the survival selection step the genotypes with higher fitness have higher probability of survival while lower quality individuals still have a small chance for survival otherwise, the evolution will turn too greedy.

The cycles of parent selection, offspring generation and mutation and the fitness evaluation will continue until a predefined termination criteria is reached. The termination criteria is often chosen to be a certain number of evolutionary cycles, or a quality threshold for the genotypes of usage of a certain amount of computational resources. Once the termination criteria is reached, the evolutionary process will stop and encode subset of the individuals with the highest quality to corresponding phenotypes and finally the phenotypes are returned as the outcome of the evolutionary optimization.

EAs are often referred to as black box optimization algorithms since they do not use any kind of domain knowledge for a given problem. The operators are defined independently of the problem: only the evaluation of the fitness function has to be implemented as long as the problem can be encoded as an unconstrained problem or as an unconstrained problem on continuous variables. However, in many applications either implicit or explicit constraints are involved, which requires the definition of problem-dependent variation operators. A relevant example of this case is the RNA inverse folding problem which requires both the recombination and mutation operators to follow a set of hard constraints.

The computational complexity of EAs is generally dominated by the computational cost associated with fitness evaluation and can be defined by $\mathcal{O}(IPn^C)$ where $I$ is the number of the evolutionary steps (iterations), $P$ is the population size, $n$ is the length of the genotype and $C$ is a positive exponent. Here $n^C$ represents the computational complexity of computing the fitness of a given solution. When fitness evaluation is a not an expensive task then EAs are very useful as they have been shown to be capable of generating high quality solutions in a relatively small number of iterations. However as the cost of fitness evaluation goes up, such as the case of RNA folding for pseudoknotted secondary structures, the EAs do not scale well.

EAs have been successfully used to solve a variety of optimization problems such as multi-objective optimization (Coello et al., 2007), parameter optimization (Michalewicz and Schoenauer, 1996), electromagnetic optimization (Rahmat-Samii and Michielssen, 1999), scheduling (Cheng et al., 1996), simulating RNA folding pathways (Gultyaev et al., 1995), consensus RNA secondary structure prediction (Chen et al., 2000), RNA inverse folding (Taneda, 2012), protein docking modelling (Gardiner et al., 2001) and protein structure prediction (Custódio et al., 2014).

### 3.6.2 Ant colony algorithms

Swarm intelligence (Engelbrecht, 2006; Kennedy, 2006) is an approach inspired from the nature and the social behaviour of insects and another animals. Several different flavours of optimization algorithms using swarm intelligence have been developed and applied to different COPs over the past three decades (Beni and Wang, 1993; Ducatelle et al., 2010; Karaboga and Akay, 2009; Karaboga and Basturk, 2007; Krishnanand and Ghose, 2009). Ant colony optimization (ACO) takes inspiration from from the foraging behaviour of some ant species. Some families of ants called Argentine ants deposit pheromones on the ground in order to mark some favourable path to a food source which should be followed by other members of their colony. Assuming a single source for food and multiple paths leading to it all paths have initially equal probability to be chosen by the ants. Ants choose a random path and when they find the food source, they leave pheromone on their way back. Shorter paths will accumulate pheromones more quickly than the longer paths. The higher the amount of pheromone in a given path, the higher the chances that the next ant chooses that path which again contributes to the accumulation of more pheromone on that path. Over time the shorter paths will accumulate more pheromone and therefore more ants will choose that path. This behaviour of ants has been the source of inspiration for solving optimization problems. In ACO a number of artificial ants build solutions to an optimization problem and exchange information on the quality of these solutions by adopting a communication mechanism similar to the one adopted by real ants.

Lets consider the TSP problem. Using ACO, the TSP is tackled by simulating a number of artificial ants moving on a graph that encodes the TSP: each vertex of the graph represents a city and each edge represents a path connecting the two cities. A variable called pheromone is associated with each edge and can be read and modified by other ants. The ACO is an iterative algorithm. At each iteration the algorithm considers a number of artificial ants. Each ant walks through the graph and the path it travelrs builds a solution. As required by the problem statement, ants can not visit each vertex more than once. This process is called solution construction. Ants select the next vertices according to a stochastic process which is biased by the pheromone value of the edge which connects the current vertex to the next vertex. When on vertex $i$, the probability of choosing edge $(i, j)$ is proportional to the amount of pheromone associated with the edge. When each ant reaches a local minima, which in the case of TSP means all possible next steps lead to a vertex which has already been visited by that ant, the iteration for that ant stops. It is a common technique to apply a local search heuristic at this stage to improve on the quality of the solution before moving to the next iteration.

Once the above iteration is finished, the quality of each solution generated by each individual ant is evaluated (i.e, similar to fitness evaluation in EAs) and the pheromone values are updated according to the quality of each solution. Once the pheromones are updated another round of iteration will start. The iterative cycle continues until a stop criteria is reached. The way the stop criteria is defined in AOC algorithms is in principle similar to the way the stop criteria is defined in EAs. Figure 11 illustrates the key algorithmic steps of a generic ACO algorithm.

The computational complexity of generic ACO algorithms remains similar to that of EAs and can be described by $\mathcal{O}(IPn^C)$ where $I$ is the number of iterations, $P$ is the population size and $n$ is the size of the problem. In the case of TSP $n$ is the number of vertices of the graph. While ACP algorithms remain an attractive and powerful method to solve COPs, one must keep in mind that if the fitness evaluation is not linear such as the case for pseudoknotted RNA folding, the ACO method does not scale well.

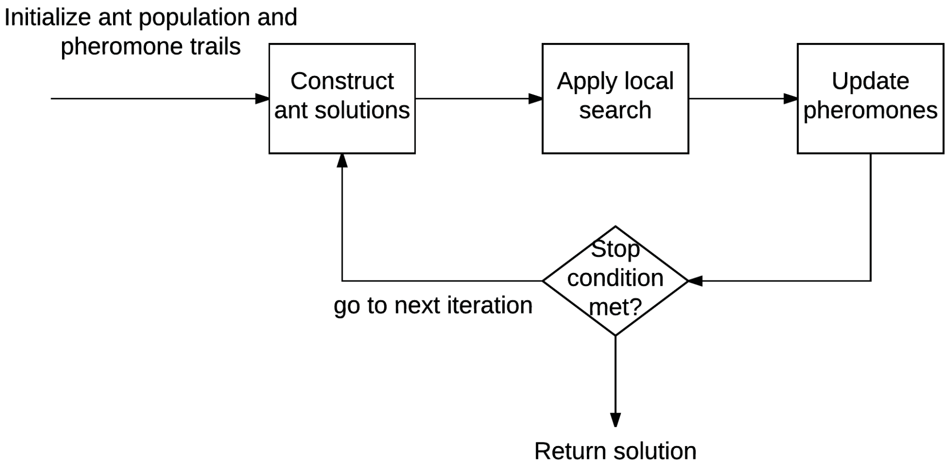


Figure 11: General schema of an Ant Colony Optimization (ACO) algorithm

Over the past three decades, several different flavours of ACO have such as Ant System (AS) (Dorigo et al., 1996), Elitist AS (Bullnheimer et al., 1997), max-min AS (Stützle and Hoos, 2000), hyper-cube AS (Blum and Dorigo, 2004) have been developed and used to successfully solve some of the very complex NP hard problems. ACO have been used to solve (Dorigo and Gambardella, 1997), vehicle routing (Bell and McMullen, 2004), graph colouring (Dorigo and Di Caro, 1999), open shop scheduling (Blum and Sampels, 2004), protein folding (Shmygelska and Hoos, 2005), drug design (Korb et al., 2006), DNA design (Kurniawan et al., 2008) and RNA design (Kleinkauf et al., 2015).

### 3.6.3 Artificial neural networks

Artificial neural networks (ANNs) were initially developed for classification, pattern recognition and function approximation problems (Cybenko, 1989; Pao, 1989; Sietsma and Dow, 1991). Later progress in design of ANNs included specialized designs which could also be used for COPs (Smith, 1999). ANNs consist of two key units named neurones and synapses. Neurones are responsible to process information and the synapses link the neurones together with some connection quality or weight. Many ANNs also have a local updating rule which determines the state of a given neurone relative to the state of the other neurones within a given vicinity.

There are two main classes of ANNs namely feed-forward networks and feed-back networks. Feedback networks are often called recurrent networks. In feed-forward ANNs (Svozil et al., 1997), the connection between the units do not form cycles and information flows in one direction from layer to layer. The information enters the network via the input nodes and then passes through a number of hidden nodes and finally exits the network through the output nodes. Feed-forward neural networks are mostly used for supervised learning cases where a labeled train data set exists and is used to train the network (Reed and Marks, 1999). In feed-back or recurrent networks (Williams and Zipser, 1989), the network connections contain directed cycles. Unlike feed-forward networks, in recurrent networks the information travels in loops from layer to layer so that the state of the network is influenced by its previous states. While feedforward neural networks can be thought of as stateless, recurrent networks have a memory which allows them to store information about its past computations. This allows recurrent neural networks to exhibit dynamic temporal behaviour. Because recurrent networks can capture temporality, they have shown to be very powerful in complex tasks such as natural language processing (NLP) (Graves et al., 2013).

ANNs have been used to solve several classes of COPs and other optimization problems including TSP (Mulder and Wunsch, 2003; Reinelt, 1994), data mining (Lu et al., 1996; Wu et al., 2014), graph colouring (Takefuji and Lee, 1991), channel assignment for cellular radio (Kunz, 1991), protein secondary structure prediction (Pollastri et al., 2002; Rost and Sander, 1994; Shen and Bax, 2013) RNA sequence analysis (Dobin et al., 2013), RNA-DNA binding (Alipanahi et al., 2015), and medical diagnosis (Amato et al., 2013). To the best of our knowledge, ANNs have not been used to solve the RNA design problem and the applicability of ANNs for the design of RNA secondary structures remains an open question.

## 3.7 Heuristics

Heuristics stand for strategies using readily accessible information to define problem-solving algorithms. Heuristics can be divided into two categories: construction heuristics and improvement heuristics. Construction heuristics construct feasible solutions for a given optimization problem from scratch. Examples of construction heuristics include nearest neighbour heuristics (Cheung and Fu, 1998), insertion heuristics (Campbell and Savelsbergh, 2004) and greedy heuristics (Chvatal, 1979). Improvement heuristics take feasible solution as input and try to improve on its quality in iterative steps. Examples of improvement heuristics neighbourhood search heuristics such as local search heuristics (Korupolu et al., 2000), simulated annealing (Kirkpatrick et al., 1983) and tabu search (Glover and Laguna, 2013).

In the following, $f(s \in S)$ is referred to as an objective function of a maximization problem $P$.

### 3.7.1 Greedy search heuristics

Greedy algorithms are constructive heuristics to build high quality solutions for COPs by making the most favourable choices at each iteration of the algorithm. The choice at each iteration is highly depending on the type of the problem as well as the state of the problem instance. Notably the choice at each stage is influenced by the choices already made in previous stages and also will influence the choice which will be made during the next stages. Greedy choices can be viewed as local decision rules and usually lead to sub-optimum solutions since the resulting solution at the end of construction is unknown and future decision may have a large impact on the resulting objective value of the solution.

Greedy search heuristics have been used to solve diverse range of problems such as minimum spanning trees (Held and Karp, 1970; Kritikos and Ioannou, 2017), routing problems (Waxman, 1988), clustering and location problems (Kazakovtsev and Antamoshkin, 2014), attribute selection (Freitag, 2017), maximum clique (Rossi et al., 2014) image segmentation (Felzenszwalb and Huttenlocher, 2004) and RNA inverse folding (Hofacker, 2003).

### 3.7.2 Local search heuristics

Local search algorithms are a family of neighbourhood search algorithms. First we present a definition for the term neighbourhood followed by the definition of local search.

A neighbourhood of a solution $s$ to a COP $P$ is defined as a set of can solutions which can be reached by applying an elementary operator $M : S \longrightarrow P(S)$ to the solution $s$. This set or the neighbourhood can be denoted by

$$\aleph(s) = M(s) \subset S \tag{15}$$

The elementary move operator $M$ describes a move set as is yields a move $m \in M$ from a feasible solution $s_i$ to another feasible solution $s_j$. Variants of local search methods use the notion of move set as a way access the nearby vicinity of each candidate solution to COPs.

Local search heuristics start with a feasible solution and via iterative greedy search of the neighbourhood generate better solutions until a local optima is found. Consider a maximization problem. Beginning from a point —which is often chosen randomly —a new solution with a higher objective function $f$ is searched in the neighbourhood. If a better solution is found, the solution is accepted as a new solution and then the neighbourhood search happens in its neighbourhood. The iterative search process continues until a local optimum is found. The effectiveness of local search depends on the choice of the neighbourhood $\aleph$. The greater the neighbourhood, the higher the chances of finding better results however enlarging the neighbourhood may lead to intractability.

One advantage of local search over other heuristics is that the search space can be searched very efficiently. A disadvantage of local search is that the solutions found are only local optima. RNAinverse from the Vienna package uses local search to efficiently find high quality solutions for the RNA inverse folding problem.

### 3.7.3 Simulated annealing and threshold accepting

Simulated annealing and Threshold accepting are two variants of the local search method. These two methods are very similar, however, they differ in that each accepts a neighbouring solution.

Local search method only accepts better solutions than the existing one, however, both simulated annealing and threshold accepting allow accepting solutions that are worse than the existing solution. Simple local search and threshold accepting methods are deterministic algorithms as for a given problem, they always produce the same solution. However, simulated annealing is stochastic (non-deterministic) method as it may produce different solutions for the same problem each time it attempts to solve it.

**Simulated annealing** method was first inspired by a technique from metallurgy involving heating and controlled cooling of a solid to obtain low energy states in a heat bath (Van Laarhoven and Aarts, 1987). The idea is to first increase the temperature of the bath to a maximum value causing the solid to melt which causes its particles to randomly distribute in the medium. Then the cooling process starts: the bath is carefully cooled down until the particles of the melted solid reach the ground state of the solid thus forming a highly structured material with minimum energy and maximum stability.

Metropolis proposed the Monte Carlo method (Metropolis and Ulam, 1949) which can be used to simulate the simulated annealing process in a computer program (Rubinstein and Kroese, 2016). Given a current state $i$ of the system with energy $E_i$, one can apply a controlled distortion mechanism to transform the state of the system to a neighbouring state $j$. The metropolis criterion describes whether the state $j$ is going to be accepted or not by applying the following acceptance logic: Let energy difference $\Delta E$ be defined by $\Delta E = E_j - E_i$. If $\Delta E \leq 0$ then accept state $j$ otherwise accept state $j$ with probability $e^{\Delta E/kT}$ where $T$ is the temperature in Kelvin and $k$ is the Boltzmann constant (Fellmuth et al., 2006). In the context of COPs, the energy function is often replaced by a cost or objective function $f$. Algorithm 1 describes a general template for the simulated annealing process.

**Algorithm 1** *simulated_annealing*$(s \in S) : s$

```
$t = T(0), n = 1$
$s_{best} = s$
while termination criteria not reached do
    $s' = generate\_neighbour(s)$
    generate a feasible neighbour
    $\Delta f = f(s) - f(s')$
    if $(\Delta f \leq 0)$ or $(e^{\Delta f/t} > random[0, 1))$ then
        $s = s'$
    end if
    if $f_s > f_{s_{best}}$ then
        $s_{best} = s$
    end if
    $t = T(n)$
    $n = n + 1$
end while
return $s_{best}$
```

The simulated annealing method has been used to assess signals in RNA sequences (Gibbs et al., 2000), describe RNA folding pathways (Schmitz and Steger, 1996), solve optimization problems related to energy conservation (Ekren and Ekren, 2010), planning and scheduling problems (Li and McMahon, 2007) and localization in wireless sensor networks (Kannan et al., 2005) to name a few.

**Threshold accepting** has been proposed as a computationally less expensive alternative to the simulated annealing method (Dueck and Scheuer, 1990). In threshold accepting the Metropolis criterion has been replaced by a simpler criterion which accepts a neighbouring solution if if the fitness value of current state $i$ and the next state $j$ differ bellow a threshold $\Theta$. It follows that unlike the simulated annealing, the threshold accepting method is a deterministic optimization technique.

The threshold accepting method has been used to solve the TSP (Reinelt, 1994), protein structure analysis (Leutner et al., 1998), scheduling (Marimuthu et al., 2009) and design of reinforced concrete bridge frames (Perea et al., 2008) to name a few.

### 3.7.4 Tabu search

Tabu search extends the simple neighbourhood search by making use of memory (Glover and Laguna, 2013). Similar to simulated annealing and threshold accepting, the accepting criterion of tabu search allows acceptance of solutions with lower fitness compared to the other immediate neighbouring solutions. However the next state must have better fitness than the current state. Tabu search uses a memory of search history to avoid getting stuck in loops or local minima. Tabu search stores a previously evaluated solution in the memory in a list named tabu list to avoid unnecessary reevaluation. As the size of the memory grows larger, the maintenance of the tabu list could become inefficient. Therefore various techniques have been used to deal with this issue. For instance the size of the memory could be limited. Algorithm 2 presents a high level template of a generic tabu search strategy.

**Algorithm 2** *tabu_search*$(s \in S) : s$

```
$t = \{\}$
$s_{best} = s$
while termination criteria not reached do
    Find best solution $s' \in \aleph(s)$ such that $s' \notin T$
    $s = s'$
    $T = T \bigcup s$
    if $f(s) > f(s_{best})$ then
        $s_{best} = s$
    end if
end while
return $s_{best}$
```

Tabu search has been successfully used to solve the vehicle routing problem (Montané and Galvao, 2006), container loading problem (Gendreau et al., 2006), jigsaw puzzles (Hoff and Olver, 2014), RNA tertiary structure prediction (Blazewicz et al., 2005), computing folding pathways between RNA secondary structures (Dotu et al., 2009) and NMR protein structure analysis (Çavuşlar et al., 2012) to name a few.

## 3.8 Stochastic search

In deterministic local search methods, one assumes that there exists perfect information about the fitness behaviour of the cost function over the fitness landscape and that the context of the solution space remains constant during movement from one position to another within the search space. However, such assumption is not always useful. In contrast, in stochastic methods the decision for the direction of movement of the algorithm within the search space at each iteration is made by using some probability distribution (Bodini and Ponty, 2010; Roussel and Soria, 2009). Algorithm 3 shows a general schema for search and optimization methods using stochastic sampling.

**Algorithm 3** $stochastic\_search(s \in S) : s$

```
1: $s_{best} = s$
2: while termination criteria not reached do
3:    $s = sample\_from\_neighbourhood(s, distance\_from\_s, sampling\_distribution)$
4:    $fitness = fitness(s)$
5:    if $fitness > fitness(s_{best})$ then
6:       $s_{best} = s$
7:    end if
8: end while
9: return $s_{best}$
```

Various sampling distributions methods such as Gibbs sampling (Ishwaran and James, 2001), Boltzmann sampling (Flajolet et al., 2007) and importance sampling (Kanj et al., 2006) have been used to solve COPs such as detection of over-presented motifs in the upstream regions of coexpressed genes (Thijs et al., 2002), clustering microarray DNA data (Sheng et al., 2003), free energy calculation of molecules (Chipot and Pohorille, 2007), random generation of combinatorial structure (Duchon et al., 2004), RNA secondary structure prediction (Ding et al., 2005) and designing RNA secondary structures with targeted nucleotide distribution (Reinharz et al., 2013) to name a few.

Notice on line 3 of algorithm 3 where the sampling from the neighbourhood (i.e, mutational landscape) of the current solution $s$ occurs; there is a parameter called $distance\_from\_s$. This parameter specifies the number of changes to be applied on a given data point in order to generate a new solution candidate within the neighbourhood of $s$. In most above mentioned methods the value for $distance\_from\_s$ is set to 1. In some sampling methods such as (Waldispühl et al., 2008) or tabu search strategies such as (Busch and Backofen, 2006) the value used for $distance\_from\_s$

maybe chosen to be greater than 1. However to the best of our knowledge most sampling methods use a fixed value for $distance\_from\_s$. An interesting avenue to explore in the context of designing RNAs could be to allow $distance\_from\_s$ to change as the properties of the search neighbour changes. One could hypothesize variable $distance\_from\_s$ may lead to reduction in number of the times the fitness evaluation function is called. This can be useful specially for cases where fitness evaluation is computationaly expensive. Variable distance between current candidate solution and the next candidate solution may result in faster convergence to high quality solutions in far regions of the mutational landscape of $s$. To the best of our knowledge such *adaptive* search strategy has never been used in the context of designing RNA structures. In a following chapter we show that in the context of RNA structure design, an adaptive stochastic search strategy leads to fast convergence and provide empirical data that this indeed is the case.

## 3.9 Memetic algorithms

It has been shown combining some form of problem domain knowledge in EAs or other search heuristics can be a highly effective optimization strategy (Burke and Silva, 2005; Ochoa et al., 2012). For instance, it has been shown that combining problem specific knowledge with EAs can increase the efficiency and effectiveness of EAs (Freisleben and Merz, 1996).

### 3.9.1 Combining evolutionary algorithms with local search methods

Local search strategies can easily allow one to embed problem specific knowledge in the neighbourhood search process. Domain specific knowledge can be used to guide the local search prociess. Thus leading to more effective neighbourhood search and therefore faster convergence such as in (Wagstaff et al., 2001).

The performance of local search strategies is highly dependent on the quality of the starting position in the search space. A starting closely positioned near a deep local minima can greatly effect the performance of the optimization process or even lead to sub-optimal solutions that are of low quality and far in distance from the global minima. Hence careful choice of the starting point is critical (Johnson et al., 1988). In contrast to local search methods, population based methods such as EAs have a smaller chance of getting trapped in local minima as they start with a population of starting points spread over different regions of the search space.

Combining EAs with a local search strategy which is guided by some form of problem specific knowledge is an effective optimization strategy (Grosan and Abraham, 2007; Oh et al., 2004). In this approach the role of the variation operators (cross-ver and mutation) can change. for instance, instead of randomly combining individuals to produce offsprings, the combination operator requires to perform a guided combination of individuals following certain rules as realized from domain specific instructions.

### 3.9.2 Memetic algorithms

Meme is the abbreviation for the Greek term mimeme and can be thought of as a unit of transmission of ideas, ways of thinking and styles that can spread within a culture. A family of biologically inspired optimization algorithms called Memetic Algorithms (MAs) (Moscato and Cotta, 2002) follow principles similar to those of EAs however with a fundamental difference: MAs are more goal oriented than the EAs where the passage of information from one generation to another happens more consciously. MA design goals can be described as following:

- Fast convergence: it is desired to find good quality solutions in fewer number of generations
- Goal oriented: individual candidate solutions cooperate with other candidates
- Diversification of the population: it is desired to develop techniques to such as neighbourhood search or tailored recombination methods to guarantee certain level of diversity in the population pool

Algorithm 4 describes a generic schema of an MA. Heuristics are used to generate a good quality initial population. All individuals in the population represent a local optima in a neighbourhood of their close vicinity. To guarantee local optimality of each member of the population, local search (Krasnogor and Smith, 2000) or tabu search (Moscato and Cotta, 2002) or simulated annealing (Knowles and Corne, 2000) algorithms are applied after the initial population is generated and also after applying the evolutionary operations. In each generation, diversification methods are applied to maintain diversity of the population (Sörensen and Sevaux, 2006).

Different flavours of MAs have been used to solve COPs such as the TSP (Moscato and Norman, 1992), inferencing gene regulatory networks (Spieth et al., 2004) and RNA phylogenetic reconstruction with maximum parsimony (Richer et al., 2009) to name a few.

**Algorithm 4** *memetic_algorithm*$(n, I)$ //n=population size, I=problem instance

```
$P = \{\}$
**for** $i$ in $range(1, n)$ **do**
    //initialize good quality population using domain knowledge
    $i = use\_heuristic\_generate\_seed(I)$
    $i = local\_search(i)$
    $P = P \bigcup i$
**end for**
**while** termination criteria not reached **do**
    **for** $x$ in $xrange(1, max\_number\_recombinations)$ **do**
        $i_a, i_b = randomly\_choose\_parents(P)$
        $i_c = dp\_recombination(i_a, i_b)$
        $i_c = do\_local\_search(i_c)$
        $P = \bigcup i_c$
    **end for**
    **for** $x$ in $xrange(1, max\_number\_mutations)$ **do**
        $i = randomly\_choose\_individual(P)$
        $i = mutate(i)$
        $i = local\_search(i)$ //or any variation such as tabu search or simulated annealing
        $P = i \bigcup P$
    **end for**
    $P = sample\_subset(P)$ // some form of sampling with bias towards diverse characteristics
**end while**
return $P$
```

# Chapter 4

# RNA Secondary Structure Design Algorithms

## 4.1 Preface

In this chapter we present a comprehensive review of the existing computational methods for the design of RNA secondary structures.

## 4.2 The State of the art in RNA design

RNA design or the inverse RNA folding problem is a special instance of the COPs and is shown to be NP-hard (Schnall-Levin et al., 2008) and most existing algorithms resort to heuristics and local search strategies to solve the problem. A general strategy is to first initialize a random seed sequence and then iteratively mutate it until the desired structural properties have emerged as predicted by folding algorithms. The choice of objective function as well as the requirement to realize some structural properties such as pseudoknots or other motifs can have significant effects on the complexity of computing the fitness function as well as the performance of the underlying optimization algorithms. The following is a comprehensive survey of the state of the art of computational methods for the design of RNA secondary structures.

`RNAinverse` (Hofacker et al., 1994) is the first and one of the most widely used RNA secondary structure design programs. Given a target secondary structure without pseudoknots `RNAinverse` attempts to find an RNA sequence by an adaptive local walk, or greedy algorithm that is predicted by `RNAfold` to have the target as its MFE structure. The initial seed sequence is randomly chosen; then sequence positions are iteratively and randomly mutated and mutations are accepted if the objective function improves. In the case of `RNAinverse`, the objective function reflects the Hamming distance between the predicted MFE structure of the design candidate and the target secondary structure. The optimization procedure stops if and when the Hamming distance reaches zero.

We note that there is no guarantee for the optimization procedure to find an optimal solution and therefore it is required to specify a limit for the maximum number of iterations allowed. `RNAinverse` is downloadable as part of the Vienna RNA package and is also available as a web service.

`RNA-SSD` (Andronescu et al., 2004) introduces a clever hierarchical decomposition approach. `RNA-SSD` constructs a random but compatible initial candidate sequence $\phi$ for the entire given RNA structure $\tau$ without pseudoknots, then structure $\tau$ and the initial sequence are hierarchically decomposed into smaller components corresponding to substructures of $\tau$. At the lowest level, these subproblems are independently solved using a conventional stochastic local search algorithm. Here the objective function is to find subsequences that have their corresponding substructures as their MFE. Solutions to these subproblems are then combined into candidate solutions for larger subproblems. There is no guarantee that valid solutions to subproblems can be combined into a valid solution of a larger subproblem; hence, at this stage, each combination attempt has to be evaluated using `RNAfold`. If at any stage the respective combined candidate sequence does not fold into the required structure, new candidate solutions to the subproblems are determined using the same mechanism as described before. Following this approach, the expensive evaluation of candidate solution happens primarily at the level of substructures which can be made small enough (by iterated decomposition) to render the $\mathcal{O}(n^3)$ complexity of `RNAfold` for larger sequences manageable. Larger candidate sequences are only evaluated after merging partial solutions, a process that happens less frequently. The hierarchical decomposition approach is based on the intuition that although there can be complicated dependencies between subproblems, there is a chance that solutions to subproblems can be successfully combined into solutions of the entire problem. The empirical performance of `RNA-SSD` on biological and artificial RNA structures supports this intuition. `RNA-SSD` outperformed `RNAinverse` in rate of success when benchmarked in multiple trials. `RNA-SSD` is available for download and also as a web service. We have not seen any reports in the literature about the quality of the sequences designed by `RNA-SSD` in any biological context.

`INFO-RNA` (Busch and Backofen, 2006) takes a similar approach to `RNA-SSD` however with an improved seed initialization method as well as a modified stochastic local search strategy. `INFO-RNA` first decomposes the input pseudoknot-free target into a set of SSEs and then follows a dynamic programming technique to find sequences that can realize each individual SSE with the minimum possible energy. Then the small sequences are assembled in a hierarchical manner similar to that of `RNA-SSD` where a stochastic local search with 10 step look ahead algorithm iteratively mutates the joined subsequences. The objective function here is to minimize the distance between the MFE structure of the joined subsequences as predicted by `RNAfold` with each corresponding sub-structure of the target structure. The final goal is to find a sequence with its MFE structure being equivalent to the target secondary structure. In benchmark studies `INFO-RNA` has shown a higher success rate than `RNAinverse` and `RNA-SSD` and has generated sequences that are thermodynamically more stable however with a bias towards having higher GC content in the composition of the generated sequences. `INFO-RNA` is available as a web service.

NUPACK-design (Zadeh et al., 2011a) introduces the notion of ensemble defect optimization where the term ensemble defect quantifies the predicted ensemble average of incorrect base pairs and free nucleotides at thermodynamic equilibrium. Given a target secondary structure without pseudoknots, NUPACK-design initializes a random seed sequence and uses the NUPACK folding algorithm to compute the partition function and the ensemble defect of the seed sequence folding into the target structure. NUPACK-design, similar to RNA-SSD, follows a hierarchical decomposition approach to decompose the target structure and the corresponding seed sequence into smaller modules. The optimization process occurs from the bottom to the top of the decomposition tree through stochastic local search by iteratively mutating single base pairs or single nucleotides at each terminal node. The mutation operator samples from the low ensemble defect mutational landscape of the immediate neighbourhood of each subsequence. Each subsequence is accepted if the ensemble defect value falls bellow a predefined threshold and the accepted subsequences are merged towards the root of the tree. The optimization process continues until a solution at the top of the decomposition tree with ensemble defect value bellow a desired threshold has emerged. NUPACK-design has been extensively utilized to design RNA nano shapes in-vitro and has shown significantly superior performance compared to RNAinverse, RNA-SSD and INFO-RNA. NUPACK-design is part of the NUPACK design and analytics package and is available in both executable as well as a well web service. There are numerous reports in the literature that reflect on the high quality of nano structures designed by NUPACK-design (Garcia-Martin et al., 2013).

Frnakenstein (Lyngsø et al., 2012) takes an evolutionary optimization approach and implements a GA to solve the RNA inverse folding where one or more target secondary structures without pseudoknots can be realized in a single run. The ability to design for multiple targets is particularly useful when designing multi-state RNAs such as riboswitches. Here the main deviation from a completely generic GA is that the method is aware of the aim of designing sequences folding into one or more target structures. Rather than searching the full sequence space, Frnakenstein directs the search by ensuring all sequences can fold into all target structures via formation of only canonical base pairs. This can be viewed as a similar approach to the local search as implemented by other methods such as RNA-SSD or INFO-RNA, except that the recombination operation chooses a random decomposition and assesses two complementary structural components in conjunction, rather than independently. The seed initialization step for generating the initial population is either via random generation of seed sequences or by running RNAinverse to design high quality seed sequences for one of the target structures. At each iteration of the GA, recombination and mutation operators seek to conserve or recombine those positions of the candidate sequences that show higher fitness than the other positions. The fitness of each position is determined by the Boltzmann probability of that position forming the correct base pair as characterized by the partition function as computed by RNAfold. The optimization criteria is to improve on the Boltzmann probability of the sequence candidates folding into the target structure(s). Frnakenstein outperforms RNAinverse, RNA-SSD and INFO-RNA in designing sequences that fold into target structure(s) with higher Boltzmann probability and comparable performance with NUPACK-design. The source code of Frnakenstein is available for download. To the best of our knowledge, there is no report in the literature to reflect on the quality of the sequences designed by `Frnakenstein` in any biological context.

`RNA-ensign` (Levin et al.) introduces the notion of global sampling to design RNA secondary structures without pseudoknots. First, `RNA-ensign` initializes a random seed sequence and then follows a sampling strategy from the low energy ensemble of the seed sequence. The low energy ensemble of the seed sequence is characterized by a set of sequences that can fold into the target structure each with a certain probability value as characterized by the partition function. Using the `RNAmutants` algorithm (Waldispühl et al., 2008) `RNA-ensign` can sample sequences from the low-energy ensemble of a given sequence such that each sample is a *k-mutant* of the input sequence where $k$ is the exact number of positions mutated at each iteration. A notable benefit of the global sampling approach where at each iteration up to $k$ positions are subject to mutation is that the sampled sequences will gradually move away from the initial seed sequence and therefore reduce the bias of designing sequences that are closely similar to the initial seed. Here the choice of $k$ is arbitrary and decided by the user. It remains unclear how $k$ should be chosen for optimal outcome. A down side of this sampling approach is the prohibitive run time requirement of `RNAmutant` which scales in $\mathcal{O}(n^5)$ time and $\mathcal{O}(n^3)$ space. On a small benchmark dataset consisting of natural and artificial RNA secondary structures of size below 100 nucleotides, `RNA-ensign` has been shown to have higher success rate than `RNAinverse` and `NUPACK-design`. However, it has been shown the sequences generated by `NUPACK-design` tend to have a higher probability of folding into the corresponding target structures. The source code of `RNA-ensign` is available for download. To the best of our knowledge, the quality of the sequences designed by `RNA-ensign` have not been verified in any biological context.

`INV` (Gao et al., 2010) is the first reported algorithm which can deal with 3-noncrossing, canonical pseudoknot structures. The term *k-noncrossing* was first introduced by the `Cross` folding algorithm (Huang et al., 2009). `Cross` MFE, 3-noncrossing RNA structures, i.e, structures that do not contain three or more mutually crossing arcs and in which each stack has size equal or greater than three. In particular, in a 3-canonical structure there are no isolated arcs. `INV` utilizes a structural decomposition technique to break the 3-noncrossing input structure into smaller 3-noncrossing modules. Then for each module, `INV` uses a generic stochastic local search to find a sequence which folds into the corresponding module as predicted by `Cross`. Then similar to `RNA-SSD`, the smaller sequences are merged to generate solution for the initial input structure. At this point judging the quality of the sequences that can be designed by `INV` as well as the performance of the algorithmic approach remains unclear. In particular, the benchmark dataset of the original paper contains only four structures of size 40-76 nucleotides and neither the source code nor any executable are available for further benchmarking purposes.

`MODENA` (Taneda, 2012) implements a bi-objective NSGA2 (Deb et al., 2000) genetic algorithm to design secondary structures including pseudoknots. At each iteration, specialized crossover operators that can handle pseudoknots and random point mutations are used to generate new offsprings.

Then MODEA uses IPknot as the folding algorithm to measure the fitness of each individual sequence in the population pool. At each iteration of the genetic algorithm, the fitness of each candidate sequence is evaluated by i) measuring the similarity between the predicted secondary structure and the target structure and ii) the free energy of the structure as predicted by IPknot. The objective of the algorithm is to minimize the free energy of each sequence when folding into the target structure, and also to maximize the structural similarity between the predicted fold of each sequence with the target structure. In the end of each iteration, the non-dominant pareto frontier of each generation is selected as the input population for the next generation. In benchmark studies MODENA's performance was compared with INV and the results greatly favoured to MODENA. To the best of our knowledge, there are biological studies to verify the quality of the sequences designed by it. MODENA is available as an executable for download.

RNAiFOLD (Garcia-Martin et al., 2013) uses a Constraint Programming (CP) approach to solve the RNA inverse folding for secondary structures. Given a target RNA secondary structure without pseudoknots, RNAiFOLD finds an RNA sequence whose minimum free energy structure is equivalent to the target structure. Similar to most of the previous methods, RNAiFOLD enables the user to specify a set of design constraints such as partial nucleotide composition for specific regions. Moreover, for the first time RNAiFOLD allows users to specify a desired level of GC content as a design constraint. RNAiFOLD performs an exhaustive exploration of the search space which can lead, in some cases, especially when the structures are large and complex, to prohibitive inverse folding times. For this reason, RNAiFOLD utilizes a Large Neighbourhood Search (LNS) method which builds on the underlying CP framework, achieving better results for larger structures. The benchmarking dataset of RNAiFOLD shows superior performance compared to RNAinverse, RNA-SSD, INFO-RNA and MODENA, but inferior performance when compared to NUPACK-design. RNAiFOLD is available as executable as well as a web service (Garcia-Martin et al., 2013). RNAiFOLD has been used to design pseudoknot-free hammerhead ribozymes and successfully verify the catalytic activity both in *in-vitro* and *in-vivo* (Dotu et al., 2014).

IncaRNAtion (Reinharz et al., 2013) implements a hybrid approach which consists of a global sampling strategy that is conceptually similar to that of RNA-ensign followed by a local search step, to design RNA structures with targeted GC content and without pseudoknots. Given a target structure, IncaRNAtion first utilizes a simplified energy model to sample sequences with highly stable stacking regions from the low energy ensemble of all sequences that can fold into the target. Notably the sampling stage does not require an initial seed sequence and therefore is considered to be seedless. This seedless approach eliminates the bias that is introduced by the initial seed. In the second stage the output sequence from the global sampling step is used as input sequence for RNAinverse for further refinement. During the local optimization step, the only nucleotides subject to mutation are those that are unpaired. The global-local approach of IncaRNAtion has shown superior performance compared to RNAinverse and RNA-SSD. To the best of our knowledge there are no studies about the quality of the sequences designed by IncaRNAtion in any biological context. The source code of IncaRNAtion is available for download.

RNAfbinv (Weinbrand et al., 2013) performs sequence design that conforms to the shape of the input pseudoknot-free secondary structure, the specified thermodynamic stability, the specified mutational robustness and the user-selected fragment after shape decomposition. In this shape-based design approach, specific RNA structural motifs with known biological functions are strictly enforced, while others can possess more flexibility in their structure in favour of preserving physical attributes. The algorithm consists of initialization of a seed which contains the desired motif sequence followed by a simulated annealing with a four nucleotide look-ahead local search function. In the first step, RNAinverse is used to design a high quality seed sequence including the desired sequence constraints. In the second step iterative mutations are performed to search for local minima and a simulated annealing approach with a non-exhaustive four nucleotide look-ahead local search function is used to sample from the vicinity of the sequence. Here the objective function seeks to minimize a weighted sum of five terms, namely: 1) structural similarity of the designed sequences as predicted by RNAfold to the target structure, 2) presence of desired motif sequence, 3) tree edit distance with the target, 4) the mutational robustness and 5) thermodynamic stability. RNAfbinv has shown superior performance when compared with RNAinverse, RNA-SSD and INFO-RNA in designing sequences under compositional constraints that are thermodynamically more stable and are also more robust to positional mutations. RNAfbinv is available as a web service.

Nanofolder (Bindewald et al., 2011) uses a simplified energy model and takes the general local stochastic local search approach that is tailored to design non-functional RNA structures that include pseudoknots. Nanofolder uses an empirical scoring function for RNA complexes rather than a physical energy function to measure the quality of the RNA sequences. The structure prediction algorithm proceeds as follows: first, an exhaustive list of all possible helices consisting of two or more base pairs is generated. Each helix is scored using the empirical scoring function. To predict an RNA secondary structure, an RNA is folded *in-silico* by placing the helices in ascending order of their scores, such that a newly placed helix is not overlapping with previously placed helices. The stochastic local search method seeks to make point mutations that are expected to improve on the probability of folding into the target structure. Notably the probability of folding into target is approximated using a simplified sampling technique that does not require computation of the partition function. The mutation operator also respects a set of design criteria that are tailored to design non functional nano structures. Nanofolder has been extensively used to design non functional nano structures in various biological contexts and is available as a web service.

AntaRNA (Kleinkauf et al., 2015) takes an energy minimization approach and an ant colony optimization (Dorigo et al., 2006) strategy to design RNA structures that include pseudoknots and have targeted GC content. AntaRNA is initialized with a random seed sequence. During the ACO process a large set of sequences are generated and finally the best solution is returned. antaRNA measures the quality of sequence candidates by computing the structural distance between the target structure and the MFE structure of each sequence candidate as predicted by pkiss, as well as the amount of the GC content. To the best of our knowledge, the quality of the sequences generated by antaRNA have not been verified in any biological context. AntaRNA is available as a web service.

ERD (Esmaili-Taheri and Ganjtabesh, 2015) implements a GA and an energy minimization criteria to design pseudoknot-free RNA structures with targeted range of free energy. First, ERD hierarchically decomposes the target structure into individual SSEs. Next a set of naturally occurring sequences from the RNA STRAND (Andronescu et al., 2008) are collected to generate a pool of seed sequences for each corresponding SSE. Using a generic bi-objective GA, ERD designs sequences with matching MFE structure to their corresponding SSEs as predicted by RNAfold, while respecting the free energy requirement. During the optimization step, the successfully designed subsequences are merged and refined until the full structure is designed. On a small benchmark dataset, ERD has shown superior performance when compared to MODENA and INFO-RNA, and comparable performance when compared to RNAiFOLD and NUPACK-design. To the best of our knowledge, the quality of the sequences generated by ERD has not been verified in any biological context. ERD is available as a web service.

Table 1 summarizes the key aspects of the methods we discussed. Note that other than Nanofolder, which is tailored for the design of non-functional nano shapes, there are no other reports on computational methods for the design of pseudoknotted RNAs where the quality of *in-silico* results have also been verified either *in-vivo* or *in-vitro*. In Chapter 5 we will introduce a novel algorithm and a design pipeline for *in-silico* design of functional RNA secondary structures including pseudoknots. In Chapter 6 we verify the applicability of our approach *in-vitro*.

| Method | Pseudoknots | Wet-lab | Optimization strategy | Optimization criteria |
|---|---|---|---|---|
| RNAinverse | ✗ | ✓ | adaptive local walk | structural distance minimization |
| RNA-SSD | ✗ | ✗ | stochastic local search | structural distance minimization |
| INFO-RNA | ✗ | ✗ | stochastic local search | structural distance minimization |
| NUPACK-design | ✗ | ✓ | weighted local sampling | ensemble defect minimization |
| Frnakenstein | ✗ | ✗ | genetic algorithm | Boltzmann probability maximization |
| RNA-ensign | ✗ | ✗ | global sampling | energy minimization |
| INV | ✓ | ✗ | graph decomposition | energy minimization |
| MODENA | ✓ | ✗ | genetic algorithm | energy and structural distance minimization |
| RNAiFOLD | ✗ | ✓ | constraint programming and stochastic local search | energy minimization |
| IncaRNAtion | ✗ | ✗ | global sampling and local optimization | energy minimization |
| RNAfbinv | ✗ | ✗ | fragment assembly and stochastic local search | weighted sum of design constraints |
| Nanofolder | ✓ | ✓ | stochastic local search | weighted sum of design constraints |
| antaRNA | ✓ | ✗ | ant colony optimization | energy minimization |
| ERD | ✗ | ✗ | fragment assembly and genetic algorithm | energy minimization |

Table 1: Summary of different computational RNA designer methods

# Chapter 5

# Enzymer: a new sequence optimization algorithm and design pipeline for pseudoknotted RNAs

## 5.1 Preface

This chapter presents our proposed design pipeline and new weighted sampling algorithm for the design and reengineering of functional RNAs including pseudoknots. This work is published in the RNA section of the journal of Frontiers in Genetics (journal impact factor 3.79) (Zandi et al., 2016). The content of this chapter presents the original publication including minor modifications and corrections compared to the published article. The key modifications include 1) the original literature review section is now more condensed , 2) an extended description of the seed initialization part of the design process as suggested by a reviewer and an updated version of Figure 12, 3) the addition of 2 new sections 5.5.2, 5.5.3) and three related extra figures (Figures 5, 19, 20) and 4) an extended discussion section.

## 5.2 Abstract

Computational design of RNA sequences that fold into targeted secondary structures has many applications in biomedicine, nanotechnology and synthetic biology. An RNA molecule is made of different types of secondary structure elements and an important RNA element named the pseudoknot plays a key role in stabilizing the functional form of the molecule. Due to the computational complexities associated with characterizing pseudoknotted RNA structures, most of the existing RNA sequence designer algorithms generally ignore this important structural element and therefore limit their applications. In this paper we present a complete design pipeline named `Enzymer` for the design of pseudoknotted RNA secondary structures. Our design methodology can leverage the

evolutionary signal found in multiple sequence alignment data to reengineer naturally occurring pseudoknotted RNAs with known function. Enzymer makes use of NUPACK as the folding algorithm to compute the equilibrium characteristics of pseudoknotted RNAs, and implements a new adaptive defect weighted sampling algorithm to design low ensemble defect RNA sequences for targeted secondary structures including pseudoknots. We used a biological data set of 201 pseudoknotted structures from the Pseudobase library to benchmark the performance of our algorithm. We compared the quality characteristics of the RNA sequences we designed by Enzymer with the results obtained from the state of the art MODENA and antaRNA. Our results show our method succeeds more frequently than MODENA and antaRNA do, and generates sequences that have a lower ensemble defect, a lower probability defect and higher thermostability. Finally by using Enzymer and constraining the design to a naturally occurring and highly conserved Hammerhead motif extracted from multiple sequence alignment data, we designed eight sequences for a pseudoknotted *cis*-acting Hammerhead ribozyme. Enzymer is available for download at https://bitbucket.org/casraz/enzymer.

## 5.3 Introduction

Ribonucleic acid (RNA) molecules play critical roles in various key cellular processes. Other than messenger RNA (mRNA) (Singer and Leder, 1966) several other classes of RNAs have been discovered to be functional and the pace of discovery has accelerated over the past decade (Fu et al., 2013; Roth et al., 2014; Stark et al., 2007; Stefani and Slack, 2008). Functional RNAs are termed non-coding RNAs (ncRNAs) because they perform their functionality directly and not via their protein products (Mattick and Makunin, 2006). NcRNAs are involved in translation (tRNA) (Giegé et al., 1993), splicing (snRNA) (Matera and Wang, 2014), processing of other RNAs (snoRNA) (Bratkovič and Rogelj, 2014) and other key regulatory processes (Bartel, 2009; Hannon, 2002; Scarborough et al., 2014; Smith et al., 2010).

Due to their diverse ranges of functionalities, ncRNAs are well suited for applications in synthetic biology (Khalil and Collins, 2010; Liang et al., 2011; Rodrigo et al., 2013), therapeutics (Burnett and Rossi, 2012; Lainé et al., 2011; Shum and Rossi, 2013), as well as nanotechnology (Afonin et al., 2013; Geary et al., 2014). The functional form of any ncRNA often requires a specific 3D structures (Shapiro et al., 2007) that is primarily determined by the secondary structure, as well as the sequence composition of the molecule (Dieterich and Stadler, 2013; Leontis and Westhof, 2003). Despite the difficulties of determining the 3D structure of RNAs, secondary structure prediction and secondary structure classification provide a major key in determining the potential functions (Laing and Schlick, 2011) as well as family signature (Griffiths-Jones et al., 2005) of the ncRNA molecules. Hence, developing better methods to design RNA sequences with specified secondary structures is a valuable pursuit as it opens doors to multiple applications.

The problem of designing artificial RNA sequences that fold into a targeted secondary structure is computationally difficult (Haleš et al., 2015; Schnall-Levin et al., 2008) and most of the existing methods resort to heuristics and stochastic local search strategies. The widely used RNA design strategy consists of two steps: first a random seed is generated; next, this seed is iteratively mutated until it adopts the desired folding properties as predicted by a folding algorithm such as `RNAfold` (Hofacker, 2003), `mfold` (Zuker, 2003) or `CentroidFold` (Hamada et al., 2009).

`RNAinverse` (Hofacker, 2003) is one of the first and most widely used RNA secondary structure design programs. `RNAinverse` decomposes the given target structure into smaller subunits and attempts to find an RNA sequence by an adaptive local walk, or greedy algorithm. After the first introduction of `RNAinverse`, significant effort by different groups has been put into the development of improved computational methods to design RNA secondary structures, many of which have shown significant improvement in the various aspects of the design process (Andronescu et al., 2004; Avihoo et al., 2011; Busch and Backofen, 2006; Garcia-Martin et al., 2013; Levin et al.; Lyngsø et al., 2012; Reinharz et al., 2013; Zadeh et al., 2011b).

All of the RNA designer methods mentioned above, ignore a critical structural element called pseudoknots and therefore have limited use. A pseudoknot is typically formed when crossing base-pairs occur between the unpaired bases from a loop and other bases outside that loop. Several ncRNA species with regulatory function such as `glmS` ribozymes (Klein and Ferré-D'Amaré, 2006; Soukup, 2006), `Delta` ribozymes (Nehdi et al., 2007), `SAM II` aptamer domain (Gilbert et al., 2008), `SAH` riboswitch aptamer domains (Edwards et al., 2010), `Hammerhead` riboyzmes (Perreault et al., 2011), the `fluoride` riboswitch (Baker et al., 2012), and `Twister` ribozymes (Roth et al., 2014) contain pseudoknots, where the pseudoknots are known to stabilize the functional form of the structure. Hence, it is of interest to develop RNA designer methods that can handle pseudoknots too. The computational complexity of designing pseduoknotted RNA secondary structures is characterized by (Ponty and Saule, 2011).

We identify three reasons why existing methods can not handle the design of pseudoknotted RNAs. First, in all of the methods the folding algorithms used to predict the folding properties of the designed sequences are often `RNAfold` or `mfold`. Even though both `RNAfold` and `mfold` can predict the MFE structure and the *partition function* (McCaskill, 1989) of a given sequence and a given target structure of length $n$ in $O(n^3)$ time and $O(n^2)$ space, neither can be used to characterize any form of pseudoknots. Second, all methods utilize hierarchical structural decomposition methods to speed up the design process. However, the hierarchical structural decomposition methods used by the previous methods can not be generalized to cover pseudoknots and therefore are inapplicable. Third, none of the methods make any distinction between the different types of base pairs (i.e, nested v.s. non-nested) and therefore are not well suited for the cases where the secondary structure includes a pseudoknot motif. In order to include pseudoknots in the design process, it is crucial to address all these shortcomings.

To our knowledge, there are three algorithmic reports in the literature for the design of pseudoknotted RNAs. `antaRNA` (Kleinkauf et al., 2015) utilizes an Ant Colony Optimization technique

(Dorigo et al., 2006) to design pseudoknotted RNAs that are predicted to fold into the target structure with targeted Guanine and Cytosine (GC) distribution. antaRNA (Kleinkauf et al., 2015) uses pKiss (Janssen and Giegerich, 2014) to predict the pseudoknotted MFE structure of RNA sequences. MODENA (Taneda, 2012) is a multi-objective genetic algorithm (MOGA) for pseudoknotted RNA sequence design. MODENA attempts to maximize the structural similarity between the target structure and the predicted fold while simultaneously minimizing the free energy of the design candidate sequences. MODENA implements a novel crossover operator to handle pseudoknots and uses IPknot (Sato et al., 2011) as its default folding algorithm. For a given RNA sequence, IPknot can predict the pseudoknotted secondary structure with *maximum expected accuracy* (MEA) (Lu et al., 2009); hence enabling MODENA to design pseudoknotted RNAs. Note that neither IPknot nor pKiss can compute the partition function and therefore can not be used to measure important qualitative characteristics such as the *ensemble defect* and the *probability defect* of the sequences. The term ensemble defect corresponds to the ensemble average of the incorrectly paired nucleotides and the term probability defect corresponds to the sum of the probabilities of all non-target structures in the structural ensemble at thermodynamic equilibrium (Zadeh et al., 2011b). INV (Gao et al., 2010) is another RNA designer algorithm to design a restricted class of pseudoknots using a graph decomposition method and a energy minimization criteria. However, as reported by (Taneda, 2012), the current implementation of INV, does not return any solution for structures larger than 85 nucleotides. It is also worth mentioning that the benchmark data set of the original article for INV, contains only four structures that are all shorter than 85 nucleotides in length.

In our work, we identify three key choices for the design of pseudoknotted RNAs and devise a pipeline which implements a novel sequence design algorithm. First is the choice of the folding algorithm which must recognize pseudoknots. Ideally, one requires the folding algorithm to compute two key measures: i) the free energy of the folded molecule, and ii) the partition function of a single RNA sequence when folded into a target pseudoknotted secondary structure. The free energy is a measure of thermostability, and the partition function makes it possible to characterize the equilibrium base pair qualities by computing the matrix of base pair probabilities. Most of the widely used single sequence folding algorithms such as RNAfold and mfold can not characterize pseudoknots. On the other hand, other existing methods, which can recognize pseudoknots such as IPknot, Hotknot (Ren et al., 2005), ProbKnot (Bellaousov and Mathews, 2010), pKiss and NanoFolder (Bindewald et al., 2011), can only compute the free energy of the pseudoknotted structures and do not make it possible to compute the partition function. To our knowledge, NUPACK is the only available method which can be utilized to compute the partition function of a limited but biologically relevant class of pseudoknots (Dirks and Pierce, 2003) and therefore makes it possible to compute the matrix of base pair probabilities of a single sequence folding into pseudoknotted target structures. Using the matrix of base pair probabilities, one can compute two other important measures namely ensemble defect and probability defect as well.

The second sequence design choice is the choice of an objective function for the optimization algorithm. antaRNA, MODENA and INV utilize energy minimization approaches to design RNA sequences that have the highest similarity to the target structure by favouring design candidates that have lower free energy when folded into the target. However, as described and demonstrated (Dirks et al., 2004; Zadeh et al., 2011b), ensemble defect optimization dominates both of energy minimization and probability defect minimization approaches. More precisely, ensemble defect minimization leads to design of molecules with folding energies that are at least as low as those of the molecules designed by energy minimization approaches and also have probability defect values that are at least as low as those of the molecules designed through probability defect minimization methods. Hence, the ideal choice for the objective function would be the ensemble defect minimization and (Zadeh et al., 2011b) provide sufficient evidence to support this claim. Notably our results verify the dominance of the ensemble defect optimization approach versus energy minimization and probability defect minimization approached.

The third sequence design choice is an efficient search and optimization strategy which may be realized via iterative sequence mutations. It is desirable for the mutation operators to be able to make a distinction between different types of base pairs (i.e., nested base pairs and non-nested base pairs), while efficiently exploring the mutational landscape of the design candidates. To efficiently explore the mutational landscape of the design candidates, the mutation operator must make effective use of the folding attributes such as the free energy, as well as the two different matrices of base pair probabilities as predicted by the folding algorithm.

In this paper, we follow an ensemble defect optimization strategy to design RNA sequences that fold into a single targeted secondary structure that includes pseudoknots. Our method extends the approach previously introduced by (Zadeh et al., 2011b) to design pseudoknot-free RNA secondary structures such that the pseudoknots can be handled as well. We introduce a complete RNA design pipeline named Enzymer which implements a new *adaptive defect weighted sampling* algorithm, and use it to progressively mutate design candidates until the specified stop conditions are reached. When available, Enzymer leverages multiple sequence alignment data to extract catalytic core of functional RNAs to facilitate seed initialization process for the purpose of reengineering ncRNAs such as ribozymes. We note that the notion of adaptive weighted sampling technique was previously used by (Reinharz et al., 2013) in another context. To benchmark our method, we used a biological dataset from the PseudoBase library (Van Batenburg et al., 2000), containing 201 pseudoknotted ncRNAs of length 21-144 nucleotides. We compared our results with the results generated by the state of the art namely MODENA and antaRNA. Our results also show that Enzymer generates sequence populations that have a lower ensemble defect, a lower probability defect, higher thermostability, a higher Boltzmann frequency and a higher success rate when compared to the results generated by antaRNA. Finally, we used the complete Enzymer pipeline and constrained the design process by using a naturally occurring and highly conserved Hammerhead motif, which was extracted from multiple sequence alignment data, to design eight RNA sequences for a pseudoknotted *cis*-acting

Hammerhead ribozyme.

## 5.4 Materials & methods

### 5.4.1 RNA folding measures at equilibrium

Let $\phi$ denote an RNA sequence with $n$ nucleotides. Sequence $\phi = \phi_1 ... \phi_n$, can be specified by positional base identities such that $\phi_i \in \{A, U, G, C\}$ for $i = 1, ..., n$. Secondary structure $\tau$ can be specified by a set of base pairs $(\phi_i, \phi_j)$ where $1 \leq i < j \leq n$, such that positions $i$ and $j$ are paired, $j \geq i + 3$, and $(\phi_i, \phi_j) \in \{(A - U), (G - C), (G - U), (U - A), (C - G), (U - G)\}$. We denote ensemble $\Gamma$, as the set of all possible secondary structures of $\phi$ including pseudoknots. For a sequence $\phi$ and secondary structure $\tau \in \Gamma$, the *free energy* $\Delta G(\phi, \tau)$ in kcal/mol, is calculated using nearest-neighbour empirical parameters for RNA in 1 M $Na^+$ (Mathews et al., 1999). By calculating the *partition function* (Dirks and Pierce, 2003) over $\Gamma$ (equation 4) one can compute the equilibrium probability of $\phi$ folding into $\tau$ (equation 5). The equilibrium structural features of ensemble $\Gamma$ are quantified by the *base pairing probability matrix* $P(\phi)$ with entries $P_{i,j} \in [0, 1]$ corresponding to the probability:

$$P_{i,j}\left(\phi\right) = \sum_{\tau \in \Gamma} p\left(\phi, \tau\right) S_{i,j}\left(\tau\right) \tag{16}$$

that the base pair $i.j$ forms at equilibrium. Here $S(\tau)$ is the *structure matrix* with entries $S_{i,j} \in \{0, 1\}$. If structure $\tau$ contains pair $i.j$, then $S_{i,j} = 1$, otherwise $S_{i,j} = 0$. To describe unpaired bases, the structure and probability matrices are augmented by an extra column. The entry $S_{i,n+1}(\tau)$ is unity if base $i$ is unpaired in structure $\tau$ and zero otherwise. The entry $P_{i,n+1}(\phi) \in [0, 1]$ denotes the equilibrium probability that base $i$ is unpaired over ensemble $\Gamma$. Hence, the row sums of the augmented $S(\tau)$ and $P(\phi)$ are unity. The term *probability defect* (Zadeh et al., 2011b) corresponding to the sum of the probabilities of all non-target structures of ensemble $\Gamma$ can be computed by term:

$$\pi(\phi, \tau) = 1 - p(\phi, \tau) \tag{17}$$

The term *ensemble defect* (Zadeh et al., 2011b) is defined by:

$$n(\phi, \tau) = n - \sum_{1 \leq i \leq n, 1 \leq j \leq n+1} P_{i,j}\left(\phi\right) S_{i,j}\left(\tau\right) \tag{18}$$

where $n(\phi, \tau)$ corresponds to the ensemble average number of incorrectly paired nucleotides at equilibrium over ensemble $\Gamma$. Intuitively, the term *normalized ensemble defect* is given by:

$$N(\phi, \tau) = n(\phi, \tau)/n \tag{19}$$

We use `NUPACK` to compute $P_{i,j}$ and two extra matrices: the matrix of nested base-pair probabilities $P'_{i,j}$, and the matrix of non-nested base-pair probabilities $P''_{i,j}$, all in $O(n^5)$ time and $O(n^4)$ space. The dynamic programming methods to compute $P'_{i,j}$ and $P''_{i,j}$ are described by (Dirks and

Pierce, 2003). Enzymer uses $P_{i,j}$ to compute the normalized ensemble defect, and uses $P'_{i,j}$ and $P''_{i,j}$ to guide the mutation operator.

One can formulate the *MFE defect* by term:

$$\mu(\phi, \tau) = d(MFE_{\phi}, \tau) \tag{20}$$

where $d(MFE_{\phi}, \tau)$ quantifies the hamming distance between the predicted MFE structure of $\phi$ and the target structure $\tau$. We call a design *successful* if $d(MFE_{\phi}, \tau) = 0$. Furthermore, to measure how dominant a structure is in the Boltzmann ensemble, one can compute the Boltzmann frequency by term:

$$B_f = e^{-\Delta G(\phi,\tau)/k_B T} / Q(\phi) \tag{21}$$

Finally, for a set of aligned sequences $S = \left\{\phi^1 ... \phi^l\right\}$ generated for a single target $\tau$, the term *sequence identity* (Reinharz et al., 2013) defined by:

$$S_{id} = \sum_{\phi^1, \phi^2 \in S \times S} \left( \frac{1}{\phi^1} \sum_{\phi^1_i \equiv \phi^2_i} 1 \right) \tag{22}$$

quantifies the the degree of similarity of the sequences in the corresponding set $S$. Intuitively, $S_{id}$ quantifies the diversity of a sequence population. Note that in our case all sequences designed for a given structure have equal length and therefore there are no gaps in the aligned set $S$.

### 5.4.2 Enzymer: an adaptive defect weighted sampling algorithm and RNA design pipeline

Enzymer follows an ensemble defect minimization approach and implements a new *adaptive defect weighted sampling* algorithm to design pseudoknotted RNAs with a single target secondary structure. In our context, the term *adaptive* means that the total number of positions to mutate at each iteration is dynamically chosen at the run-time. When the quality of the candidate sequences are low, more positions will be mutated, and as the quality improves the number of mutations at each iteration will become smaller. We show this strategy leads to a reduction in the number of required iterations; hence reducing the run-time required to reach the stop criterion. The term *defect weighted sampling* means that at each iteration the probability of mutation of a nucleotide at each position depends on the type of that position (i.e. free, nested pair or non-nested pair), and is also proportional to the positional contribution of that nucleotide to the ensemble defect of the sequence. The positional defect of each position is based on the type of the position and is quantified by $P_{i,j}$ for free nucleotides, by $P'_{i,j}$ for nested base pairs, and by $P''_{i,j}$ for non-nested base pairs. Enzymer may optionally use MSA data to extract highly conserved nucleotides often referred to as catalytic core of RNA structures with known function from Rfam or other literature in order generate a design template. The use of design template is particularly useful when the goal is to reengineer RNA enzymes with known function such as ribozymes and riboswitches.

For a given pseudoknotted target structure $\tau$ of size $n$, our method starts with either a randomly generated seed $\phi$ or a design template. The design template can be generated when MSA data as well as the consensus secondary structure for the target structure are available. The design template includes nucleotides that were highly conserved through the evolution and are known to encode the core function for ncRNAs. Once the seed sequence is generated and the design template is embedded inside the seed, then `Enzymer` iteratively samples from the low ensemble defect mutational landscape of the initial sequence until it reaches the stop condition. We note that the nucleotide positions that are part of the design template are immutable. Let $f_{stop}$ denote the maximum value that we accept for $N(\phi, \tau)$. The iterations stop and return $\phi$ when $N(\phi, \tau) \leq f_{stop}$. We note that during each instance of the design trial, there is no guarantee of reaching $N(\phi, \tau) \leq f_{stop}$. Hence, we limit the maximum number of the iterations and once the limit is reached, we report the fittest result that was found during the sampling process. Let $max\_it$ denote the maximum number of iterations. Then we define the *stop condition* as the event where either $N(\phi, \tau) \leq f_{stop}$ or $max\_it$ is reached.

Figure 12 presents the key steps of `Enzymer`. Algorithm 5 describes the complete design approach. Algorithms 2, 3 and 4, describe the three mutation operators that constitute the adaptive defect weighted sampling process. An `Enzymer` instance, starts with four input parameters: i) $\tau$, ii) $f_{stop}$, iii) $max\_it$, and iv) design template $t$ as defined by string $t = t_1...t_n$, where $t_i \in \{A, U, G, C, o\}$ such that the length of $t$ is equal to $n$. We use $t$ to specify design constrains.

First, for target $\tau$ we initialize a random RNA seed sequence $\phi$ that is compatible with the target structure by enforcing base pairing rules (Algorithm 5, line 3). At the seed initialization step, the design template $t$ generated by Algorithm 9 is used as a mean to specify a set of positional nucleotide constrains on the seed sequence. We use a random initial seed if no template is provided. We further describe the template initialization process in Algorithm 9. Once the seed is initialized, we update the seed to match the template such that for $i = 1...n$, if $t_i \neq$ "$o$" then $\phi_i = t_i$. Furthermore, $t$ is also used during the sampling process to safeguard the constrained positions against mutations. More precisely, for $i = 1...n$, the nucleotide $\phi_i$ is subject to mutation, if and only if $t_i =$ "$o$". Our algorithm allows the user to specify the percentage of the GC content for unconstrained regions of the initial seed sequence and if the GC content is not specified, a random value between of 20% to 80% is used to generate the initial seed sequence.

Second, we use the `prob` program with `-pseudo` option from `NUPACK`, to compute $P_{i,j}$, $P'_{i,j}$ and $P''_{i,j}$. We use $P_{i,j}$ to compute $N(\phi, \tau)$ and use $P'_{i,j}$ and $P''_{i,j}$ to guide the sampling step (Algorithm 5, lines 6 and 7).

Third, the algorithm executes the adaptive defect weighted sampling process until it reaches the stop condition (Algorithm 5, line 13). At each iteration the sampling process will uniformly and randomly select from one of the mutation operator (Algorithm 5, line 15) to sample mutants from the low ensemble defect mutational landscape of $\phi$. The first mutation operator targets unpaired positions and mutates a single unpaired position. The second mutation operator targets pair positions and mutates a single base pair. Ideally, we would like to mutate multiple positions at each iteration with the aim of reaching the stop criteria with fewer iterations and therefore reducing the running time of the sampling algorithm. Therefore we implemented a third mutation operator to dynamically decide for variable $m$, which quantifies the total number of positions that have to go under mutation at each iteration. Once the third mutation operator computes $m$ it will make random calls to the first and second mutation operators until precisely $m$ positions are mutated. The details of each of the three mutation operators follows:

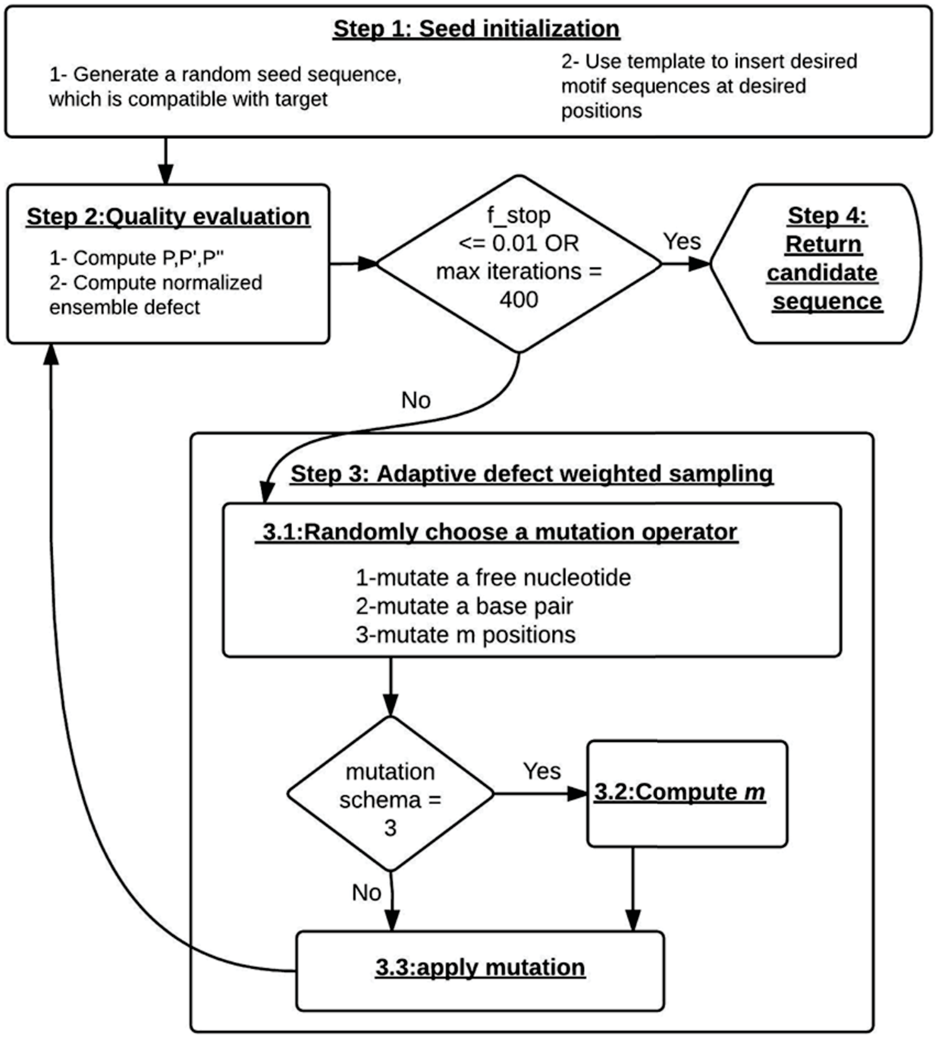

Step 1: Seed initialization
1- Generate a random seed sequence, which is compatible with target
2- Use template to insert desired motif sequences at desired positions
Step 2:Quality evaluation
1- Compute P,P',P''
2- Compute normalized ensemble defect
f_stop <= 0.01 OR max iterations = 400
Yes
Step 4: Return candidate sequence
No
Step 3: Adaptive defect weighted sampling
3.1:Randomly choose a mutation operator
1-mutate a free nucleotide
2-mutate a base pair
3-mutate m positions
mutation schema = 3
Yes
3.2:Compute m
No
3.3:apply mutation

Figure 12: Enzymer design pipeline; Step 1: generate initial seed; if a template if provided then use it as the initial seed. Step 2: evaluate the quality sequence candidates. If the stop condition is met, we return the sequence. Step 3: the adaptive defected weighted sampling process starts. In 3.1 the mutation operator is uniformly randomly selected. If the *m-mutation* schema is chosen then in step 3.2 compute the value of *m*. In 3.3 sample from low ensemble defect mutational landscape of the current sequence by applying the mutation operator; if the quality of the sampled sequence is improved compared with the , then we discard the old sequence and select the new one for further optimization. Step 4: when the stop condition is reached, return the designed sequence.

1. **single point mutation** (algorithm 6): this operator samples a mutant sequence from the mutational landscape of $\phi$ by mutating a single free nucleotide. For an arbitrary unpaired $\phi_i$, the probability of mutation is computed by $(1 - P_{i,n+1})$, which is the measure of positional contribution of $\phi_i$ to $N(\phi, \tau)$. The mutation operator scans through $\phi$ until it selects a single unpaired nucleotide $\phi_i$ for mutation.

**Algorithm 5** $Enzymer(\tau, f_{stop}, max\_it, template = False)$

```
// input: target structure, target normalized ensemble defect, maximum iterations
if ($template$ is $False$) then
    $\phi \leftarrow initialize\_random\_seed(\tau, t)$
end if
if ($template$ is not $False$) then
    $\phi = template$
end if
$iteration\_count \leftarrow 1$
$C_{design\_begin} \leftarrow current\_time()$
$P_{i,j}, P'_{i,j}, P''_{i,j}, \pi(\phi, \tau) \leftarrow nupack\_pairs(\phi, \tau)$ //compute pair probabilities using NUPACK-pairs
$N(\phi, \tau) \leftarrow compute\_normalized\_ensemble\_defect(P_{i,j}, \phi, \tau)$
// adaptive defect weighted sampling process starts here
while ($N(\phi, \tau) \geq f_{stop}$) OR ($iteration\_count < max\_it$) do
    $iteration\_count \leftarrow iteration\_count + 1$
    $mutation\_scheme \leftarrow random\_integer(1, 3)$
    if ($mutation\_scheme == 1$) then
        $\phi' \leftarrow mutate\_single\_nucleotide(\phi, \tau, P_{i,j}, t)$
    end if
    if ($mutation\_scheme == 2$) then
        $\phi' \leftarrow mutate\_basepair(\phi, \tau, P'_{i,j}, P''_{i,j}, t)$
    end if
    if ($mutation\_scheme == 3$) then
        $m' \leftarrow (length(\tau) * N(\phi, \tau))/5$
        $m \leftarrow floor(absolute\_value(normal\_distribution(m', m'/5)))$
        if $m < 1$ then
            $m = 1$
        end if
        $\phi' \leftarrow m\_mutants(m, \phi, \tau, P_{i,j}, P'_{i,j}, P''_{i,j}, t)$
    end if
    $P_{i,j}, P'_{i,j}, P''_{i,j}, \pi(\phi', \tau) \leftarrow nupack\_pairs(\phi', \tau)$
    $N(\phi', \tau) \leftarrow compute\_normalized\_ensemble\_defect(P_{i,j}, \phi', \tau)$
    if $N(\phi', \tau) < N(\phi, \tau)$ then
        $\phi = \phi'$
    end if
end while
$C_{design\_end} \leftarrow current\_time()$
$C_{design} \leftarrow C_{design\_end} - C_{design\_begin}$
Return $\phi, N(\phi, \tau), \pi(\phi, \tau), \Delta G(\phi, \tau), C_{design}$
```

2. **pair mutation** (algorithm 7): this operator samples a mutant sequence from the mutational landscape of $\phi$ by mutating a single base pair. This operator makes distinction between the two different types of base pairs. For an arbitrary nested base pair $(\phi_i, \phi_j)$, the probability of pair mutation is proportional to its contribution to $N(\phi, \tau)$ and is computed by the term $(1 - P'_{i,j})$. For an arbitrary non-nested base pair $(\phi_i, \phi_j)$, the probability of pair mutation is proportional to its contribution to $N(\phi, \tau)$ and is computed by $(1 - P''_{i,j})$. The operator continuously scans through all base pairs to select exactly one base pair for mutation.

3. **m-mutation** (algorithm 8): this operator samples a mutant sequence from the mutational landscape of $\phi$ by mutating exactly $m$ positions. The value of $m$ will dynamically converge to a value proportional to $N(\phi, \tau)$ and $n$. Let $m'$ represent the value that $m$ converges to and be defined by:

$$m' = (N(\phi, \tau) * n)/C \tag{23}$$

   where $C$ is an arbitrary constant. In our simulations we set $C = 5$. Then we compute $m$ using:

$$m = \lfloor |normal\_distribution(m', m'/5)| \rfloor \tag{24}$$

   Once the value of $m$ is determined, the operator will iteratively make uniformly random calls to the single point and pair mutation operators until exactly $m$ positions are mutated. This technique causes the sampling process to choose more positions for mutation when $N(\phi, \tau)$ is large, and to choose fewer positions as $N(\phi, \tau)$ diminishes.

The last step of each iteration is to compute $N(\phi, \tau), P_{i,j}, P'_{i,j}, P''_{i,j}$. If the newly generated sequence has improved quality (lower defect), we discard the old sequence and replace it with the newly generated one (algorithm 5, line 32). Finally the algorithm decides whether the stop condition is reached or not. When the sampling process reaches the stop condition, the iterations will stop and $\phi$ will be returned.

### 5.4.3 Run-time requirement

To measure the run-time performance of each `Enzymer` instance, we count the number of iterations as well as the number of seconds required to reach the stop criteria. We emphasize that our algorithm utilizes `NUPACK` to compute the partition function of each sequence in $O(n^5)$ time. Due to the expensive computational costs associated with computation of the partition function at each iteration, it would be ideal to utilize an approach that enables the algorithm to reach the stop criteria in fewer steps. We will discuss in the results section how our third mutation operator (i.e. m-mutation operator) improves the run-time requirement of our adaptive weighted sampling algorithm.

**Algorithm 6** *mutate_single_nucleotide*$(\phi, \tau, P_{i,j}, t)$

```
// input: sequence, target structure, matrix of pair probabilities and the design template
$mutation \leftarrow False$
while $mutation == False$ do
  $i \leftarrow randomly\_select\_unpaired\_position(\tau)$
  if $t[i]$ is not "$o$" then
    continue
  end if
  $random\_number \leftarrow random\_float(0, 1)$
  $probability\_of\_mutation \leftarrow 1 - P_{i,n+1}$
  if $random\_number < probability\_of\_mutation$ then
    $\phi' \leftarrow mutate\_at\_position(position = i, \phi)$ //replace $\phi_i$ with A,G,U or C
    if $\phi'$ is not $\phi$ then
      $\phi \leftarrow \phi'$
      $mutation \leftarrow True$
    end if
  end if
end while
Return $\phi$
```

### 5.4.4 Dataset

To benchmark the performance of our method we use a non-redundant and diverse biological dataset of pseudoknotted secondary structures prepared by (Taneda, 2012). We note that the original source of all of the target structures in this dataset is the `Pseudobase` library. The initial dataset was composed of 266 structures. We emphasize that the only existing folding algorithm which enables one to compute $P(\phi, \tau)$, $P'(\phi, \tau)$ and $P''(\phi, \tau)$, is `NUPACK` and therefore we use it to filter the dataset. Since `NUPACK` can only recognize a limited class of pseudoknots, our filtering process yields a dataset of 201 pseudoknotted structures of length 21-144 nucleotides. Figure 13 in the supplementary material section presents the size distribution of the target structures in the filtered dataset. We will refer to the filtered dataset as `Pseudo`. Our algorithm accepts secondary structures over the alphabet $\{[, ], (, ), .\}$ presented in standard dot bracket notation. The `Pseudo` dataset is available at `https://bitbucket.org/casraz/enzymer`.

**Algorithm 7** $mutate\_basepair(\phi, \tau, P'_{i,j}, P''_{i,j}, t)$

```
//function inputs: sequences, target, nested pair probability, non-nested pair probability, template
$mutation \leftarrow False$
while $mutation == False$ do
    $i, j \leftarrow randomly\_select\_a\_pair(\tau)$
    if $t[i]$ is not "$o$" AND $t[j]$ is not "$o$" then
        continue // The entire pair is locked as specified by the design template $t$
    end if
    if $t[i]$ is not "$o$" AND $t[j]$ is "$o$" then
        $\phi' \leftarrow only\_mutate\_j(j, \phi)$ // respecting pair rules, only mutate the unlocked part of the pair
        if $\phi'$ is not $\phi$ then
            $\phi \leftarrow \phi'$
            $mutation \leftarrow True$
        end if
        Return $\phi$
    end if
    if $t[j]$ is not "$o$" AND $t[i]$ is "$o$" then
        $\phi' \leftarrow only\_mutate\_i(i, \phi)$ // respecting pair rules, only mutate the unlocked part of the pair
        if $\phi'$ is not $\phi$ then
            $\phi \leftarrow \phi'$
            $mutation \leftarrow True$
        end if
        Return $\phi$
    end if
    if $(i, j)$ is a nested base pair in $\tau$ then
        $random\_number \leftarrow random\_float(0, 1)$
        $probability\_of\_mutation \leftarrow 1 - P'_{i,j}$
        if $random\_number < probability\_of\_mutation$ then
            $\phi' \leftarrow mutate\_position\_i\_j(\phi, i, j)$ //replace $\phi_i, \phi_j$ with A-U, G-C or G-U
            if $\phi'$ is not $\phi$ then
                $\phi \leftarrow \phi'$
                $mutation \leftarrow True$
            end if
        end if
    end if
    if $(i, j)$ is a non-nested base pair in $\tau$ then
        $random\_number \leftarrow random\_float(0, 1)$
        $probability\_of\_mutation \leftarrow 1 - P''_{i,j}$
        if $random\_number < probability\_of\_mutation$ then
            $\phi' \leftarrow mutate\_position\_i\_j(\phi, i, j)$ //replace $\phi_i, \phi_j$ with A-U,G-C,G-U,U-A,C-G or U-G
            if $\phi'$ is not $\phi$ then
                $\phi \leftarrow \phi'$
                $mutation \leftarrow True$
            end if
        end if
    end if
end while
Return $\phi$
```

**Algorithm 8** $m\_mutation(m, \phi, \tau, P_{i,j}, P'_{i,j}, P''_{i,j}, t)$

```
// This function mutates exactly $m$ positions. The inputs are the number of positions to mutate, sequence, target structure, pair probabilities, nested pair probabilities, non-nested pair probabilities and the design template, respectively.
$mutation\_count \leftarrow 0$
while $mutation\_count < m$ do
    $i \leftarrow random(1, length(\tau))$
    if $\phi_i$ is a free nucleotide OR $mutation\_count == (m-1)$ then
        $\phi \leftarrow mutate\_single\_nucleotide(\phi, \tau, P_{i,j}, t)$
        $mutation\_count \leftarrow mutation\_count + 1$
    end if
    if $\phi_i$ is not a single nucleotide then
        $\phi \leftarrow mutate\_basepair(\phi, \tau, P'_{i,j}, P''_{i,j}, t)$
        $mutation\_count \leftarrow mutation\_count + 2$
    end if
end while
Return $\phi$
```

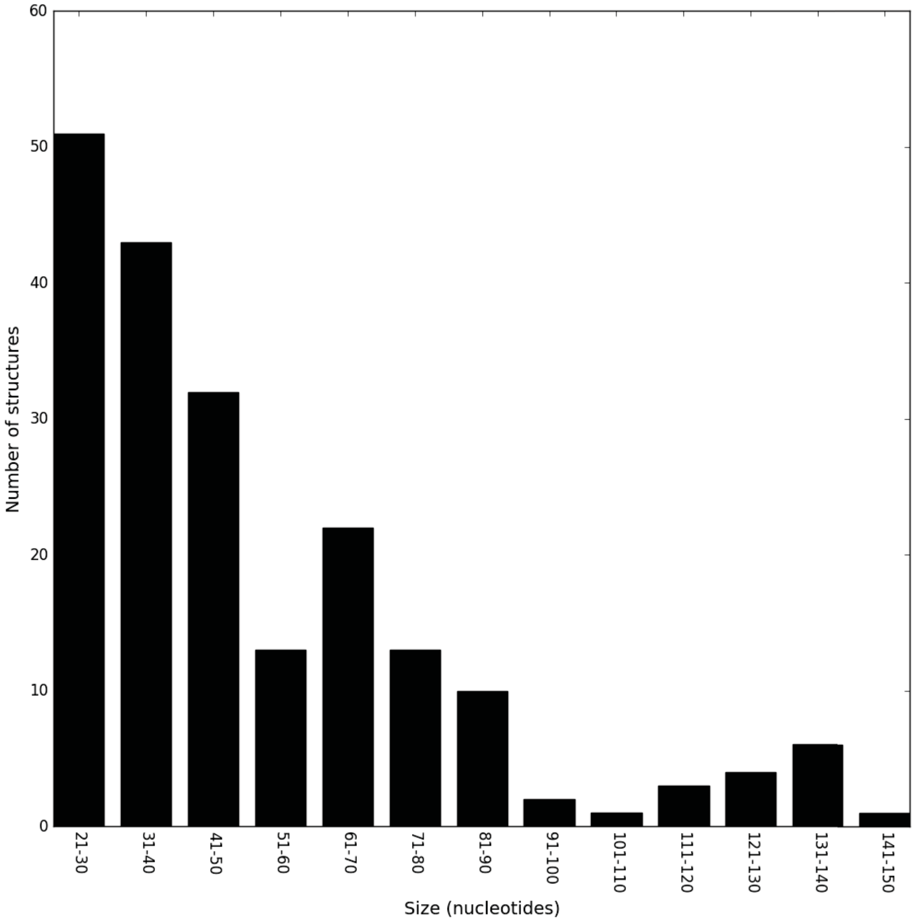


Figure 13: Benchmark dataset curated from `Pseudobase`

### 5.4.5 Setup

For each target structure in `Pseudo`, we ran `Enzyner` for 30 independent trials. We ran each trial on a dedicated computational core with a CPU speed of 2.0 GHz and 2GB of RAM. This leads to $30 * 201$ (total of 6030) independent instances of the method. In our setup, we set $f_{stop} = 0.01$ and $max\_it = 400$. Note $max\_it = 400$ is an arbitrary choice; however as we will discuss, it turned out the 400 is a sufficiently large number of iterations to demonstrate the effectiveness of our approach. Finally, `Enzymer` returns a single design candidate per trial.

For each target structure in `Pseudo`, we ran `Enzyner` for 30 independent trials. We ran each trial on a dedicated computational core with a CPU speed of 2.0 GHz and 2GB of RAM. This leads to $30 * 201$ (total of 6030) independent instances of the method. In our setup, we set $f_{stop} = 0.01$ and $max\_it = 400$. Note $max\_it = 400$ is an arbitrary choice; however as we will discuss, it turned out that 400 is a sufficiently large number of iterations to demonstrate the effectiveness of our approach. Finally, `Enzymer` returns a single design candidate per trial.

We compare the performance of `Enzymer` with `MODENA` and `antaRNA`. We emphasize that for target structure $\tau$, `Enzymer` seeks to design sequence $\phi$ by minimizing the normalized ensemble defect value, where `MODENA` and `antaRNA` aim to design sequences with high thermostability. In order to establish a fair basis for comparison with `MODENA`, we set the maximum number of generations (i.e. $max\_it$) of a `MODENA` instance to 400. Note that `MODENA` is a genetic algorithm and is initialized by a population of $P$ independently generated seed sequences and once it reaches the maximum number of generations it returns a population of $P$ candidate solutions. In order to observe a consistent behaviour, the author of `MODENA` (Taneda, 2012) recommends to set the initial population size to be equal to 10% of the total number of generations. Hence, for each target structure we set the $P = 40$. In the end, for each target structure, we sort the generated sequences based on the corresponding normalized ensemble defect values and select a subset of 30 sequences with the lowest normalized ensemble defect. `MODENA` generated sequences for all of the 201 target structures. For the case of `antaRNA`, we ran 30 independent trials and generated 30 sequences for each target structure. Because there is no guarantee that `antaRNA` reaches the stop condition, we limit the running time to be equal median running time that was required by `Enzymer` to reach the stop condition for each corresponding target structure. We note that `antaRNA` failed to recognize four of the target structures from the benchmark dataset.

Other than `MODENA` and `antaRNA`, the only other reported pseudoknot designer algorithm is `INV`. As of the date of submission of this article, `INV` has remained unavailable for benchmarking purposes. However, as reported by (Taneda, 2012), `INV` does not return any solution for structures that are larger than 85 nucleotides in length. Furthermore, even for structures that are shorter than 85 nucleotides, `MODENA` has demonstrated superior performance compared to `INV`. Therefore comparing `Enzymer` with `MODENA` and `antaRNA` is expected to provide us with sufficient information about the performance of `Enzymer`.

## 5.5 Results

### 5.5.1 Benchmark results

To characterize the quality of a designed sequence $\phi$ that is predicted to fold into $\tau$, we measure the normalized ensemble defect $N(\phi, \tau)$ (eq. 19), probability defect $\pi(\phi, \tau)$ (eq. 17), normalized free energy $\Delta G(\phi, \tau)$, MFE defect $\mu(\phi, \tau)$ (eq. 20), Boltzmann frequency $B_f$ (eq. 21) and sequence identity $S_{id}$ (eq. 22).

For each of the three methods and for each target structure $\tau^k \in$ `Pseudo` where $k = 1, ..., 201$, we generated 30 sequences $\phi^l$s where $l = 1, ..., 30$. For each $\tau^k$, let $f^k$ denote the frequency of reaching $N(\phi^l, \tau^k) \leq 0.01$ . Figure 15 presents the $f^k$ values we obtained for each $\tau^k$ from a pool of 30 generated $\phi^l$ by each method. In this performance evaluation, we observed $f^k \geq 1$ in 188, 140 and 24 cases for `Enzymer` (Figure 15 (A)), for `MODENA` (figure 15 (B)) and for `antaRNA` (Figure 15 (C)) respectively. Furthermore, we observed that there is no single case where the $f^k$ of the results generated by `Enzymer` was lower than that of `MODENA` or `antaRNA`.

The number of successful designs where $\mu(\phi^l, \tau^k) = 0$ are presented in Figure 16. The results show that `Enzymer` outperformed both `MODENA` and `antaRNA` in 191 and 194 cases respectively. We also observe `MODENA` outperformed `antaRNA` in 127 cases. Respective binomial test statistics with p-values $1.55e^{-44}$ and $1.52e^{-48}$ shows `Enzymer` delivers superior performance compared to `MODENA` and `antaRNA` in generating sequences that have their predicted MFE equal to the target structure. Moreover, binomial test statistic with p-value $2.26e^{-4}$ also shows that `MODENA` delivers superior performance compared to `antaRNA`.

Figure 17 presents the median normalized ensemble defect values of the sequences generated by each method for each target structure. We observe `Enzymer` generated sequences with lower normalized ensemble defect and outperformed both `MODENA` and `MODENA` in 200 and 201 cases respectively. Furthermore, we also observe `MODENA` outperformed `antaRNA` in 155 cases. Respective binomial test statistics with p-values $1.25e^{-58}$ and $6.22e^{-61}$ shows that `Enzymer` delivers superior performance compared to `MODENA` and `antaRNA` in generating sequences with lower ensemble defect. Furthermore, binomial test statistic with p-value $5.28e^{-15}$ shows that `MODENA` delivers superior performance as compared to `antaRNA`.

Figure 18 shows median probability defect values of the sequences generated by each method for each target structure. We observe `Enzymer` outperformed `MODENA` and `antaRNA` in 196 and 201 cases respectively. We also observe `MODENA` outperformed `antaRNA` in 153 cases. Respective binomial test statistics with p-values $1.66e^{-51}$ and $6.22e^{-61}$ shows that `Enzymer` delivers superior performance compared to `MODENA` and `antaRNA` in generating sequences with lower probability defect. Furthermore, binomial test statistic with p-value $5.72e^{-14}$ shows that `MODENA` delivers superior performance when compared to `antaRNA` as well.

Figure 19 presents the normalized median free energy values of the sequences generated by each method. We observed Enzymer designed sequences with lower free energy compared to MODENA and antaRNA in 102 and 198 cases respectively. We also observe when compared with antaRNA, MODENA generated sequences with lower free energy in 195 cases. Respective binomial test statistics with with p-value 0.88 shows Enzymer and MODENA generate sequences with similar free energy. However, respective binomial test statistics with p-values $8.42e^{-55}$ and $5.45e^{-50}$ shows that both Enzymer and MODENA deliver superior performance as compared to antaRNA in generating sequences that have lower free energy and therefore are thermodynamically more stable.

Figure 20 presents the median Boltzmann frequencies achieved by each of the methods. We observe Enzymer outperformed MODENA and antaRNA in generating sequences with higher Boltzmann frequency in 197 and 201 cases respectively. We also observe MODENA outperformed antaRNA in 153 cases. Respective binomial test statistics with p-values $4.19e^{-53}$ and $6.22e^{-61}$ shows that Enzymer delivers superior performance compared to both MODENA and antaRNA in generating sequences that have higher Boltzmann frequency values. Moreover, binomial test statistic with p-value $5.72e^{-14}$ shows that MODENA delivers superior performance compared to antaRNA.

Figure 21 presents median sequence identity for sequence populations generated by each method. We observe antaRNA generated sequences with lower sequence identity in all 201 cases. When we compare Enzymer with MODENA, we observe Enzymer generated sequences with lower sequence identity in 193 cases. Binomial test statistics with p-value $6.22e^{-61}$ suggest antaRNA generates solution sets that have lower sequence identity than those sequences generated by Enzymer and MODENA. On the other hand binomial test with p-value $3.72e^{-47}$ suggests that MODENA generates solution sets with the lower degree of sequence diversity than the solution sets generated by Enzymer.

### 5.5.2 Comparing the run-time performance of the three optimization algorithms

To compare the performance of our adaptive defect weighted sampling optimization algorithm with NSGA II of MODENA or ACO of antaRNA, we decouple the effect of the quality evaluation step which occurs at each iteration of the search and optimization process from the design process. For each method and trial we compute $R = \frac{C\_design}{C\_eval}$ where $C\_design$ represents the time in seconds to complete the design process and $C\_eval$ represents the number of seconds it takes to evaluate the quality of a single design candidate. The value of $R$ allows us to implicitly compare the relative performance of each method without requiring an explicit count of the number of times the fitness method is called. This is particularly useful since there was no clear way to count the total number of times antaRNA is calling pKiss. Figure 22 shows the $R$ values we obtained after each trial for each method. We observe that our adaptive sampling strategy shows faster convergence than NSGA II and ACO while NSGA II shows the worst performance. This observation is expected. We know the NSGA II evaluated the fitness of an entire population of solution candidates at each iteration which can be a significantly higher number of fitness evaluations compared to our approach where the fitness of only a single solution candidate is evaluated at each iteration. Similar expensive fitness evaluation schema is also observed in ACO, however, in a more moderate way as the number of individual ant agents are often smaller than the size of the population pool of genetic algorithms such as in the NSGA II. Our results show that our sampling strategy computes the fitness function only once at each iteration is very useful for problems where fitness evaluation is an expensive task. On the other hand what we also observe in Figure 22 is that our method is less sensitive than ACO and NSGA II to the size of the target structure in terms of number of fitness evaluations required while the NSGA II is highly sensitive to the size of the problem. The observation related to sensitivity of performance to size of the problem implies that adaptive weighted sampling strategies scale better than ACO and NSGA II. The results show that the adaptive weighted sampling algorithm of `Enzymer` is overall a more efficient optimization algorithm than both the ACO algorithm of `antaRNA` and the multi-objective NSGA II of `MODENA`.

Figure 23 (A) compares the run-time performance, including the time required for fitness evaluation of `Enzymer`, with `MODENA`. The y-axis quantifies the logarithm of the median running time required by each of the two methods to reach the corresponding stop criteria. The x-axis represents the size of the target structures in increasing order. As the size of the target structures grow, we observe a rapid rate of growth in the run-time requirement of `Enzymer` as opposed to a slower growth of run-time requirement for `MODENA`. The computationally costly run-time requirement of `Enzymer` is directly related to the expensive task of computing the partition function over the pseudoknotted ensemble in $O(n^5)$ time. We have omitted `antaRNA` from this figure because in our simulations we enforced `antaRNA` to run for the exact same amount of time that was required by `Enzymer` to reach the stop condition for each corresponding target structure. We note that the stop criteria for `antaRNA` is when the MFE defect becomes zero, however, as Figure 16 shows, there is no guarantee for `antaRNA` to reach the stop criteria and therefore an artificial cap on the maximum running time allowed must be applied. Figure 23 (B) presents the median value for the number of iterations required for `Enzymer` to reach the stop criteria. We observe in 179 or 89% of the cases, the stop condition was reached in less than 200 iterations. Note that given a population of size 30, the NSGA II must compute the fitness function 30 times at each iteration which means that by the time `Enzymer` reaches the stop condition, `MODENA` has only reached its 6th generation. Both `MODENA` and `antaRNA` have been omitted from Figure 23 (B). `MODENA` is omitted because it does not stop the optimization process unless it reaches the maximum number of iterations. We also omitted `antaRNA` because it was not possible to measure the total number of iterations before `antaRNA` reached the stop condition.

### 5.5.3 Effect of individual algorithmic ingredients on convergence

The effect of adding the adaptive sampling technique to the normalized ensemble defect and probability defect values are presented in Figure 24. In order to make a visual comparison possible, we also added the second degree curve to each dataset. We observe when we enabled the adaptive sampling schema (i.e., the third mutation operator) we reached the lower normalized ensemble defect values in 199 out of 201 cases (figure 24 (A)). We also observe that the adaptive sampling technique lowered the probability defect values in 181 out of 201 cases (figure 24(B)). Respective binomial test statistics with p-values $1.26e^{-56}$ and $1.25e^{-33}$ strongly suggest that when the total number of iterations are kept constant (i.e $max\_it = 400$), the adaptive sampling strategy enables the algorithm to reach a lower normalized ensemble defect value and a lower probability defect values and therefore improve on the run-time requirement of the algorithm.

Figure 25 depicts the effect of making a distinction between nested and non-nested base pairs when a paired position is subject to mutation. To understand this effect, we ran `Enzymer` in three different modes. First, `Enzymer` was ran in the default mode (red triangles in figure 25). Second `Enzymer` was ran in local search mode (one mutation per iteration) without making distinction between nested and non-nested where probability of mutation was read from the pair probability matrix $P$ (green triangle in figure 25). Third `Enzymer` was ran in local search mode but this time we allowed the sampling algorithm to make distinction between nested and non-nested pairs by reading the probability of mutation from the corresponding pair probability matrices $P'$ and $p''$ respectively (yellow stars in figure 25). A comparison of the second and third modes reveals the benefit of making a distinction between the different types of base pairs when a paired position is evaluated for mutation. More precisely, when during the sampling process with a fixed number of iterations the paired positions are mutated according to their type (nested v.s non-nested) we observe better results both in normalized ensemble defect and probability defect values. The 5th degree curves help to better compare the performances of the three modes. As expected, the first mode outperforms the other two.

### 5.5.4 Analyzing the convergence of the optimization algorithm

In principle, our optimization algorithm (Algorithm 5) takes similar steps to genetic algorithms (GA) we presented in Chapter 3 take. Our algorithm can be regarded as a GA with a population size of one without the possibility of having a cross-over operator. An advantage of our method compared with GAs is that because the population size is equal to one, our method does not require to compute the objective function more than once per iteration. As described by (Eiben et al., 1990) one can generalize the optimization strategies of GAs by following:

1. Choose an initial random point $P$ in the search space.
2. Until stop criteria is reached, find a neighbouring point $P'$ for the current candidate $P$.
3. If quality of $P'$ is better than $P$, then replace $P$ with $P'$ and go to step 2.
4. If quality of $P'$ is not better than $P$, then choose $P$ and go to step 2.

For GAs, Markov chains have been used to prove probabilistic convergence of the best solution within a population to the global optimum if at each iteration the best solution survives with probability one (Fogel, 1992; Rudolph, 1994). According to (Rudolph, 1994) convergence to the global optimum is not an inherent property of the GAs but rather is a consequence of the algorithmic trick of keeping track of the best solution found over time. It is proved by means of homogeneous finite Markov chain analysis that a GA will never converge to the global optimum. One should refer to theorem three and four on pages 4-5 from (Rudolph, 1994) for the formal proof. However, according to theorem seven in (Rudolph, 1994) on page seven, maintaining the best solution found before selection, makes the underlying Markov process inhomogeneous and guarantees convergence to the global optimum. Notably, according to the above procedure, `Enzymer` takes precisely the same steps as GAs do and therefore guaranteed to converge to the optimal solution if enough number of iterations are passed. The first step, seed initialization of the above procedure is equivalent to lines 2 to 7 of Algorithm 5. The second step of the above procedure is equivalent to the loop which starts at line 13 and ends at line 35 of Algorithm 5. The selection step (steps 3 and 4 of the above procedure) is equivalent to line 32 of Algorithm 5. The third step of the above procedure guarantees that the best solution is always surviving. Indeed in Figure 23 we observe that the probability of reaching $N(\phi, \tau)$ in less than 200 iterations is 89% that is 179 cases out of 201. Notably we do not observe correlation between the size of the problem and the number of iterations required to reach $N(\phi, \tau) \leq 0.01$.

## 5.6 Enzymer pipeline: Using naturally occurring motif sequences to reengineer functional RNAs

To reengineer naturally occurring RNAs such as ribozymes and riboswitches, we add a preprocessing step to the seed initialization step to `Enzymer` (Algorithm 9). During the preprocessing step we generate a design template as input seed sequence for `Enzymer`. For a target molecule, we obtain the multiple sequence alignment data related to that molecule from Rfam or other sources in the literature. We use the homology profiles to extract the probabilistic profile of the evolutionarily conserved catalytic core of the target functional unit. We use the obtained homology profile to generate a population of design templates (Algorithm 9 line 1). Then we evaluate the quality of each design template (Algorithm 9 line 2) . For each template, we lock the positions with 90% rate of evolutionary conservation so they can not change during the optimization process (Algorithm 9 line 3). We allow the rest of the positions (i.e., those with less than 90% conservation rate) to remain open for mutation during the optimization process. Next we evaluate the quality of each design template (Algorithm 9 line 4) and choose the best one as input to `Enzymer` (Algorithm 9 line 5). Figure 14 and Algorithm 9 summarize the above mentioned process.

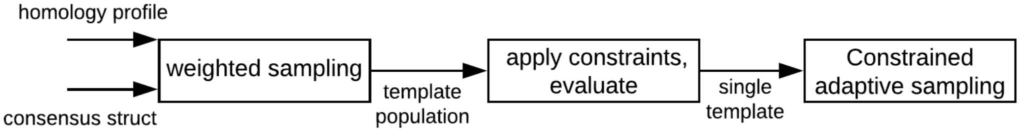


Figure 14: For a consensus secondary structure and a homology profile sample from the probability distribution of the homology profile to generate a population of ten RNA sequences. Note that 10 is an arbitrary number. Then evaluate the ensemble defect of each template in the template population. Select the best template and apply positional constraints on nucleotides with greater than or equal to 90% rate of conservation according to the homology profile. Use the final template as input to `Enzymer`.

**Algorithm 9** $Enzymer\_pipeline(homology\_profile, conserve\_rate, \tau, f\_stop, max\_it, sample\_size)$

```
$T \leftarrow sample\_from\_homology\_profile(homology\_profile, sample\_size, conserve\_rate)$ // sample_count is size of the sample set
$T \leftarrow evaluate\_ensemble\_defect(T)$
$T \leftarrow apply\_constraints(T)$
$template \leftarrow choose\_best\_template(T)$ // choose template with lowest ensemble defect
$phi \leftarrow Enzymer(\tau, f\_stop, max\_it, template = template)$
Return $\phi$
```

We used the technique described in Figure 14 to reengineer a naturally occurring hammerhead ribozyme. Hammerhead ribozymes are small self-cleaving RNAs that promote strand scission by internal phosphodiaster transfer. In this section we use `Enzymer` to reengineer a *cis*-acting pseudoknotted Hammerhead ribozyme by using a set of naturally occurring and highly conserved nucleotides, which constitute a highly conserved Hammerhead motif that is extracted from multiple sequence alignment data. An RNA structural motif is defined as a collection of nucleotides that fold into a stable three dimensional (3D) structure, which can be found in naturally occurring RNAs in unexpected abundance.

Figure 26 shows the secondary structure of a Hammerhead ribozyme from the mouse gut metagenome as reported by (Perreault et al., 2011) and we will refer to it as $HH$. The reporting article also identifies the set of highly conserved motif nucleotides with $\geq$ 90% conservation throughout the neighbouring phylogenetic family of the ribozyme. Let the design template $t_{HH}$ specify the highly conserved Hammerhead motif for the wild type $HH$. We adopt the motif specification from (Perreault et al., 2011), and generated a popupation of initial seeds. Then we evaluated the initial seeds and chose the best one and used it to it to describe the RNA template sequence for $HH$ by $t_{HH} = ooooooooooooooooCCUGAUGAGoooooooo\ oooooooGCGAAAooooooooooooooooooUCGooo$

*ooooooooooo*. We used $t_{HH}$ as the design template for `Enzymer` and use $HH$ as the target structure and designed 8 sequences $\phi^l_{HH}$ where $l = 1...8$ for the Hammerhead ribozyme. We also set $max\_it = 400$ and $f_{stop} \leq 0.01$.

Table 2 presents the quality of the sequences we generated for $HH$. The last two rows show the mean and median values of the corresponding columns. Notably $f_{stop}$ was satisfied in neither of the design trials, however, the median normalized ensemble defect achieved was as low as 0.04. Interestingly, we observed that the median value for the free energy of the designed sequences is equal to $-2.48E + 01$ which is equivalent to the free energy of the wild type sequence of the Hammerhead ribozyme. The sequences we generated are presented in Appendix A.

| Annotation | $N(\phi^l_{HH}, HH)$ | $\pi(\phi^l_{HH}, HH)$ | $\Delta G(\phi^l_{HH}, HH)$ | $max\_it$ |
|---|---|---|---|---|
| $\phi^1_{HH}$ | $4.01E - 02$ | $5.41E - 01$ | $-3.21E + 01$ | 400 |
| $\phi^2_{HH}$ | $4.97E - 02$ | $6.33E - 01$ | $-2.13E + 01$ | 400 |
| $\phi^3_{HH}$ | $5.02E - 02$ | $6.66E - 01$ | $-2.47E + 01$ | 400 |
| $\phi^4_{HH}$ | $4.34E - 02$ | $5.85E - 01$ | $-2.66E + 01$ | 400 |
| $\phi^5_{HH}$ | $4.43E - 02$ | $5.76E - 01$ | $-2.33E + 01$ | 400 |
| $\phi^6_{HH}$ | $4.99E - 02$ | $6.44E - 01$ | $-2.49E + 01$ | 400 |
| $\phi^7_{HH}$ | $4.29E - 02$ | $5.73E - 01$ | $-2.19E + 01$ | 400 |
| $\phi^8_{HH}$ | $5.38E - 02$ | $7.05E - 01$ | $-2.65E + 01$ | 400 |
| Mean | $4.68E - 02$ | $6.16E - 01$ | $-2.52E + 01$ | 400 |
| Median | $4.70E - 02$ | $6.09E - 01$ | $-2.48E + 01$ | 400 |

Table 2: The data generated for the hammerhead ribozyme

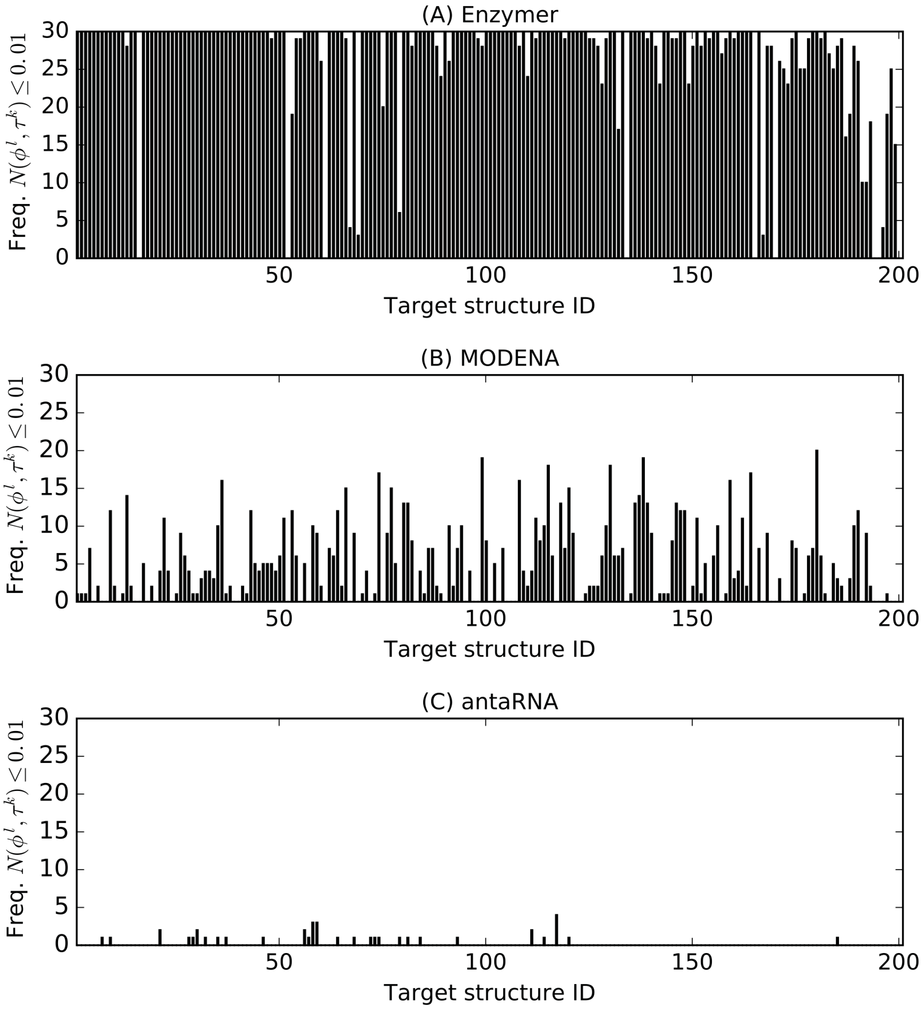


Figure 15: Frequency of solutions with low normalized ensemble defect - Frequency of the solutions per structure where $N(\phi^l, \tau^k) \leq 0.01$. For each target $\tau^k \in$ `Pseudo` for $k = 1...201$, the corresponding vertical bar represents the frequency (out of 30 trials) of the generated sequences $\phi^l$ for $l = 1...30$, where $N(\phi^l, \tau^k) \leq 0.01$. (A) `Enzymer` generated at least one sequence $\phi^l$ such that $N(\phi^l, \tau^k) \leq 0.01$ for 188 of the structures. (B) `MODENA` generated at least one sequence $\phi^l$ such that $N(\phi^l, \tau^k) \leq 0.01$ for 140 of the structures. (C) `antaRNA` generated at least one sequence $\phi^l$ such that $N(\phi^l, \tau^k) \leq 0.01$ for 24 of the structures. Binomial statistic test with 99% confidence indicates `Enzymer` significantly outperforms both `MODENA` and `antaRNA` in generating sequences such that $N(\phi^l, \tau^k) \leq 0.01$. Notably, the binomial test also indicates superior performance of `MODENA` compared with `antaRNA`. Structure IDs on the x-axis are sorted based on increasing size of the corresponding targets.

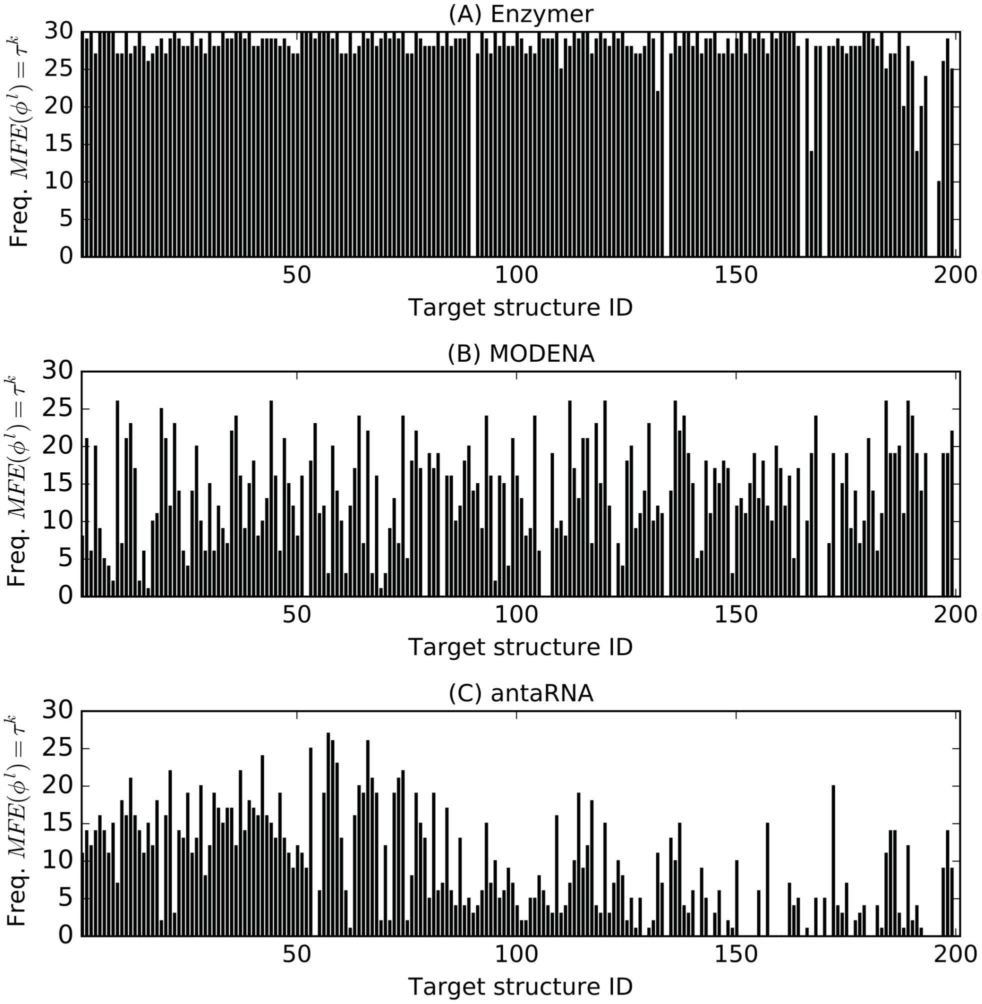


Figure 16: Frequency of successful designs - For each target $\tau^k \in$ `Pseudo` for $k = 1...201$, the corresponding vertical bar represents the frequency (out of 30 trials) where $MFE(\phi^l, \tau^k) = \tau^k$. Comparison of performance of `Enzymer` (A) with the performance of `MODENA` (B) and `antaRNA` (C) shows `Enzymer` outperformed the other two methods in 191 and 194 cases respectively. A binomial test statistic with 99% confidence indicates `Enzymer` outperforms both methods in generating sequences with lower MFE defect. Furthermore, `MODENA` outperforms `antaRNA` in 127 cases and the binomial test statistic indicates superior performance of `MODENA` compared with `antaRNA`. Structure IDs on the x-axis are sorted based on increasing size of the corresponding targets.

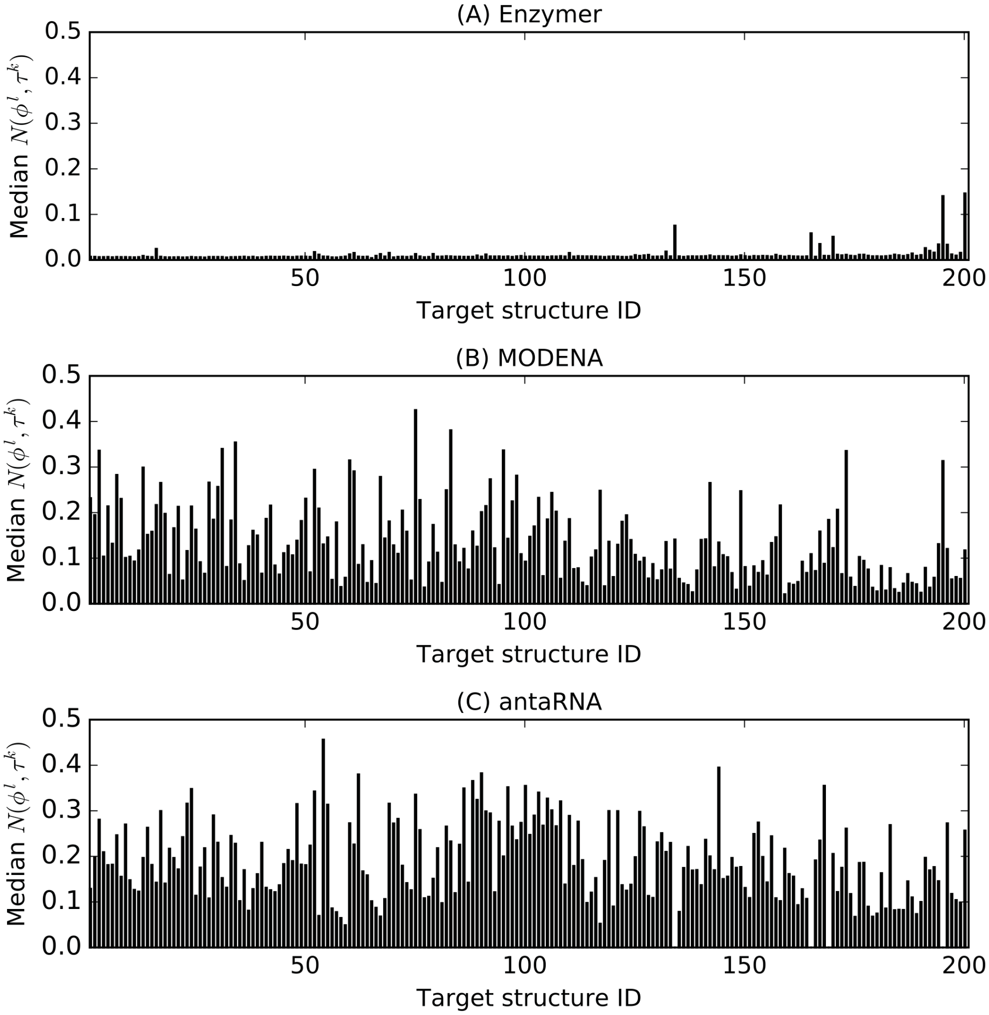


Figure 17: Distribution of normalized ensemble defect values - Comparing normalized ensemble defect. In each figure, each vertical bar represents the median $N(\phi^l, \tau^k)$ obtained for each corresponding target. The results show `Enzymer` (A) outperformed both `MODENA` (B) and `antaRNA` (C) in 200 and 201 cases respectively. A binomial test statistic with 99% confidence indicates `Enzymer` delivers significantly better results compared to the other two methods. Furthermore, `MODENA` outperformed `antaRNA` in 155 cases and the binomial test static indicates that `MODENA` delivers significant superior performance compared to `antaRNA`.

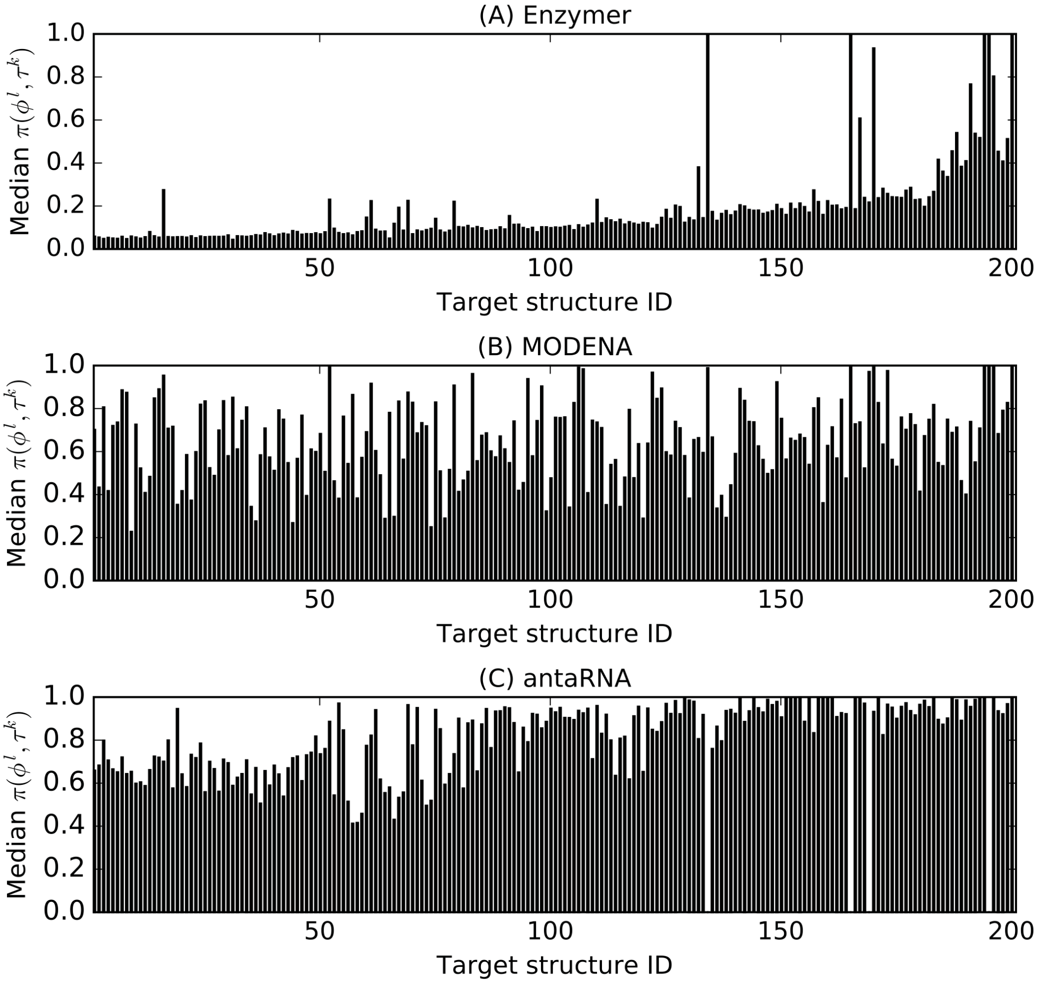


Figure 18: Comparing probability defect values - In each figure, each vertical bar represents the median $\pi(\phi^l, \tau^k)$ obtained for each corresponding target. The results show `Enzymer` (A) outperformed both `MODENA` (B) and `antaRNA` (C) in 196 and 201 cases respectively. A binomial test statistic with 99% confidence indicates `Enzymer` delivers significantly better results compared to the other two methods. Furthermore, `MODENA` outperformed `antaRNA` in 153 cases and the binomial test static indicates that `MODENA` delivers significant superior performance compared to `antaRNA`.

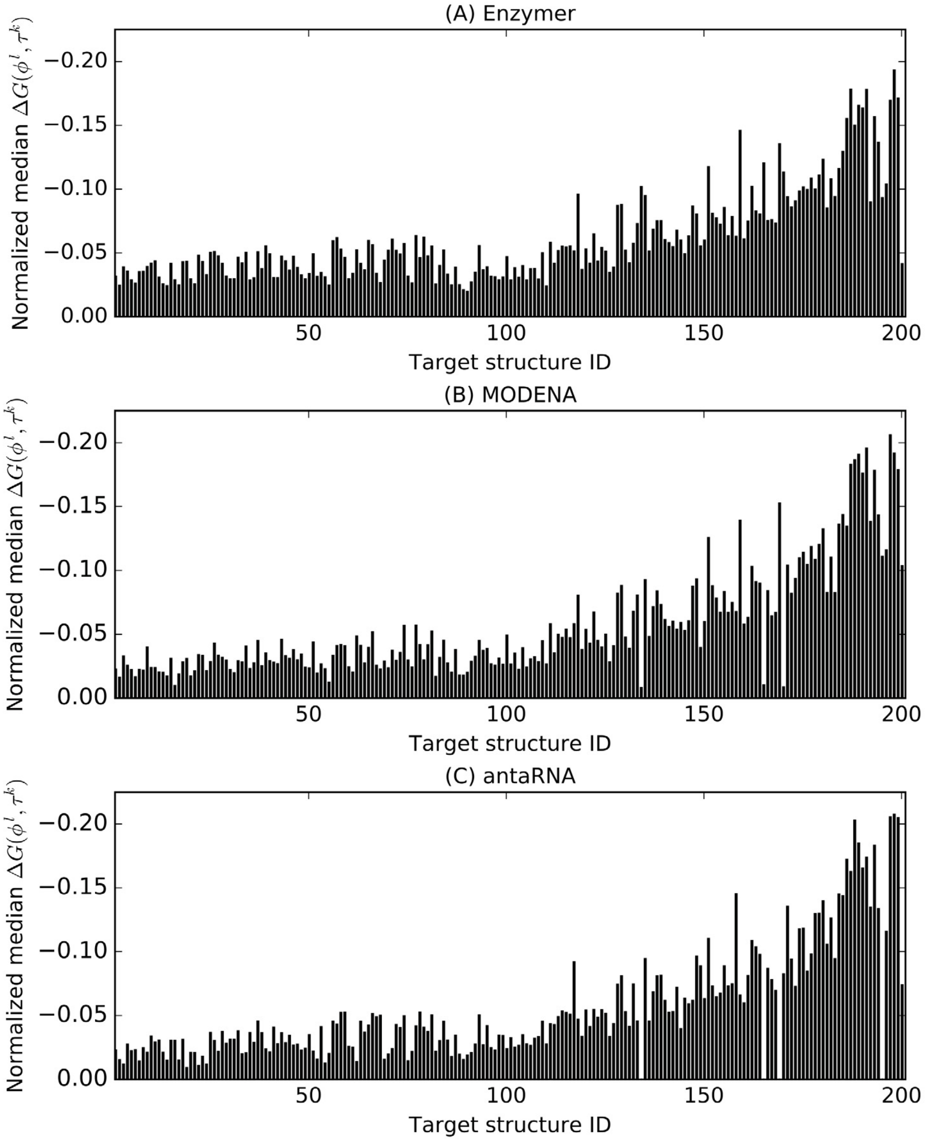


Figure 19: Comparing median normalized free energy - Comparing normalized median free energy. In each figure, each vertical bar represents the median 2 obtained for each corresponding target. The results show `Enzymer` (A) outperformed both `MODENA` (B) and `antaRNA` (C) in generating sequences with lower free energy in 102 and 198 cases respectively. A binomial test statistic with 99% confidence indicates `Enzymer` delivers significantly better results to `antaRNA`, however similar performance to `MODENA`. Furthermore, `MODENA` outperformed `antaRNA` in 195 cases and the binomial test static indicates that `MODENA` delivers significant superior performance compared to `antaRNA`.

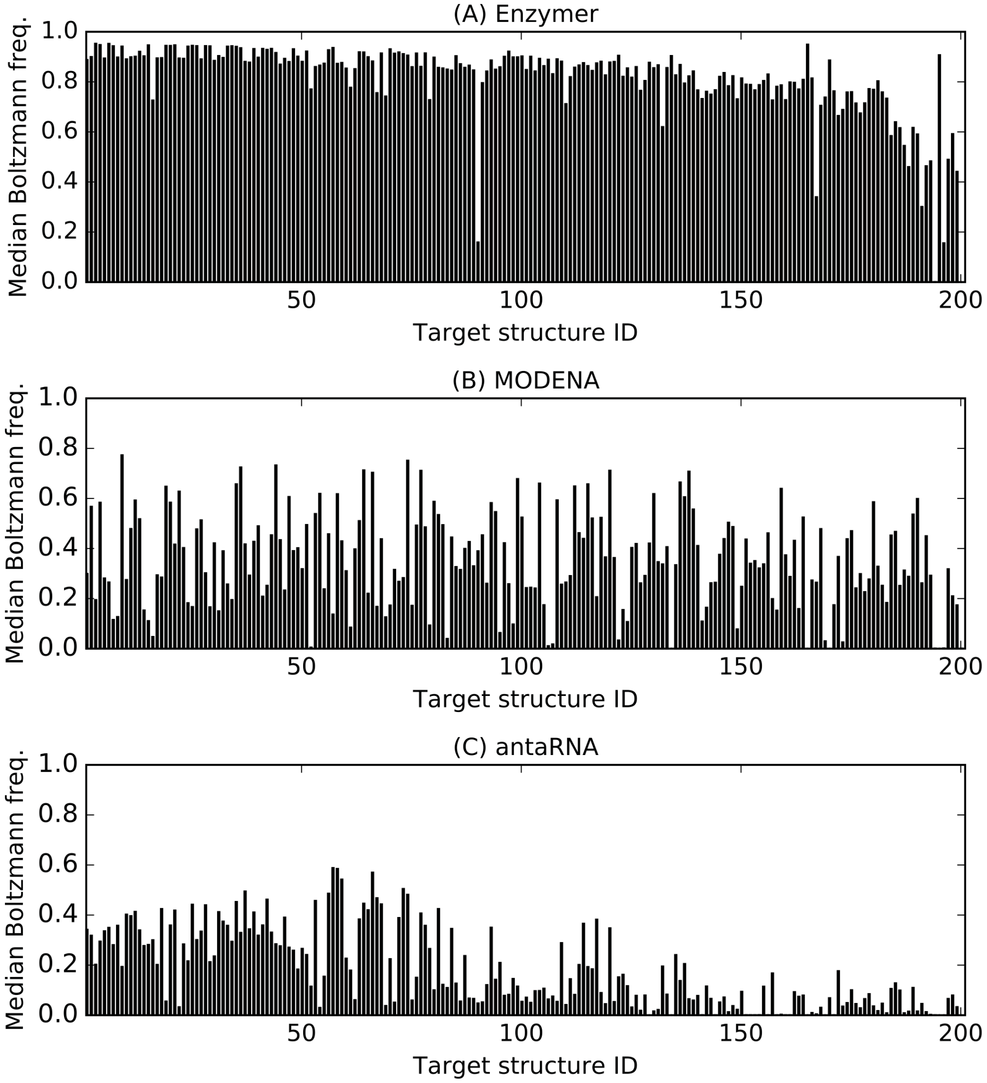


Figure 20: Comparing Boltzmann frequencies - In each figure, each vertical bar represents the median Boltzmann frequency obtained for each corresponding target. The results show `Enzymer` (A) outperformed both `MODENA` (B) and `antaRNA` (C) in 197 and 201 cases respectively. A binomial test statistic with 99% confidence indicates `Enzymer` delivers significantly better results compared to the other two methods. Furthermore, `MODENA` outperformed `antaRNA` in 153 cases and a binomial test static indicates that `MODENA` delivers significant superior performance compared to `antaRNA`.

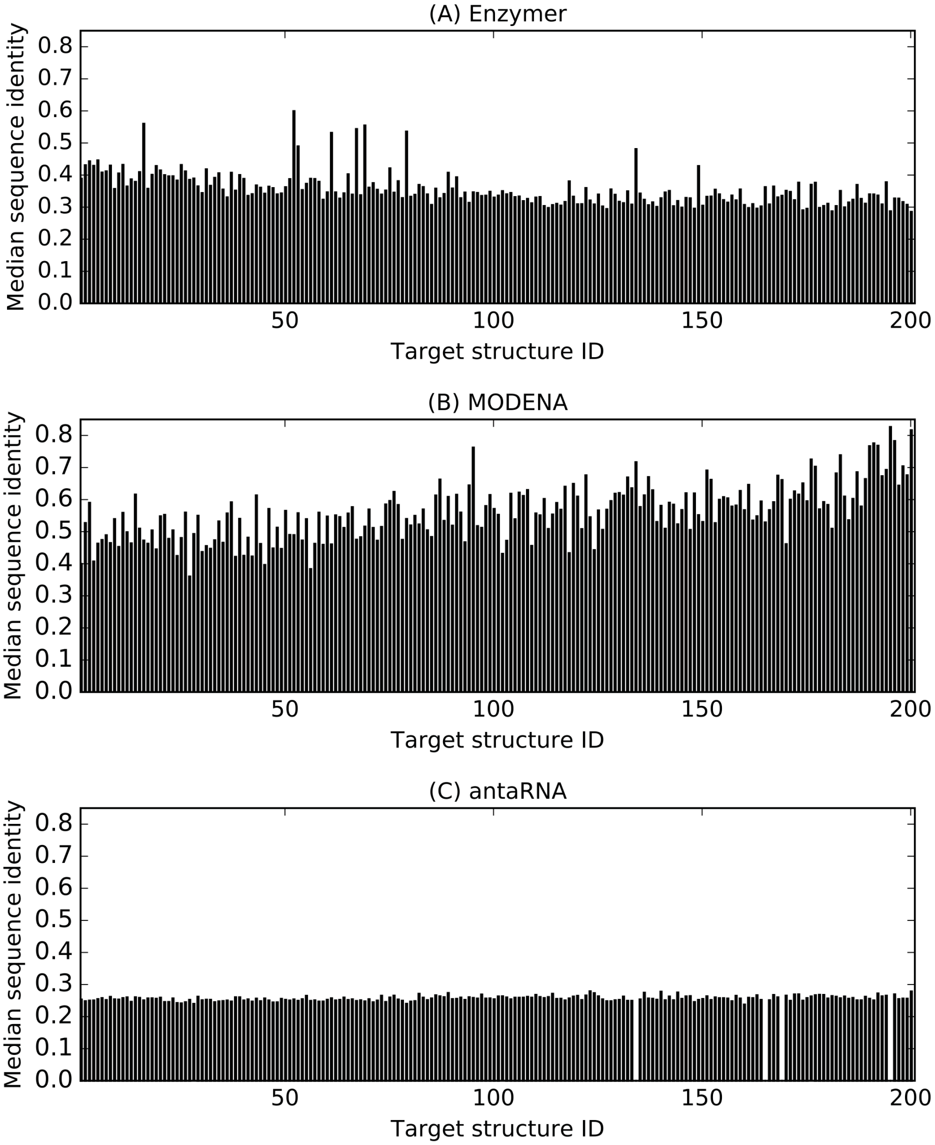


Figure 21: Comparing sequence identity - In each figure, each vertical bar represents the median sequence identity obtained for each corresponding target. For all 197 out of 201 cases where `antaRNA` (C) returned solutions, the median sequence identity was lower than `Enzymer` (A) as well as `MODENA` (B). On the other hand in 193 cases `Enzymer` generated sequences with lower sequence identity when compared with `MODENA`. Binomial test statistic with 99% confidence indicates `antaRNA` outperforms the other methods in generating sequence populations that are more diverse while `MODENA` generates sequences with the lowest sequence diversity.

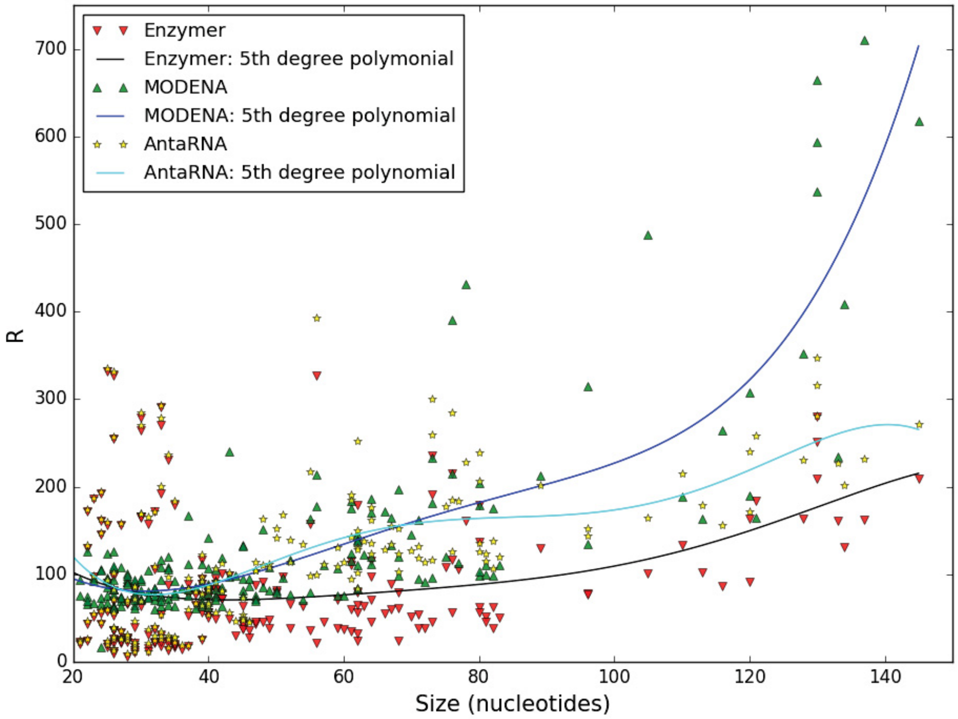


Figure 22: Comparing run-time performance of the optimization algorithms - For `antaRNA` and `MODENA` each data point represents the $R$ value obtained for each target. For `Enzymer` each data point represents the median value for $R$ obtained for each target. The data suggests the adaptive defect weighted algorithm of `Enzymer` is a more efficient optimization algorithm than the NSGA II implemented by `MODENA` and the ACO implemented by `antaRNA`. The 5th degree curves are fit to facilitate visual comparison. We iteratively fitted curves from degree 1 and above and stopped when the change in respective fitting error was smaller than 0.01.

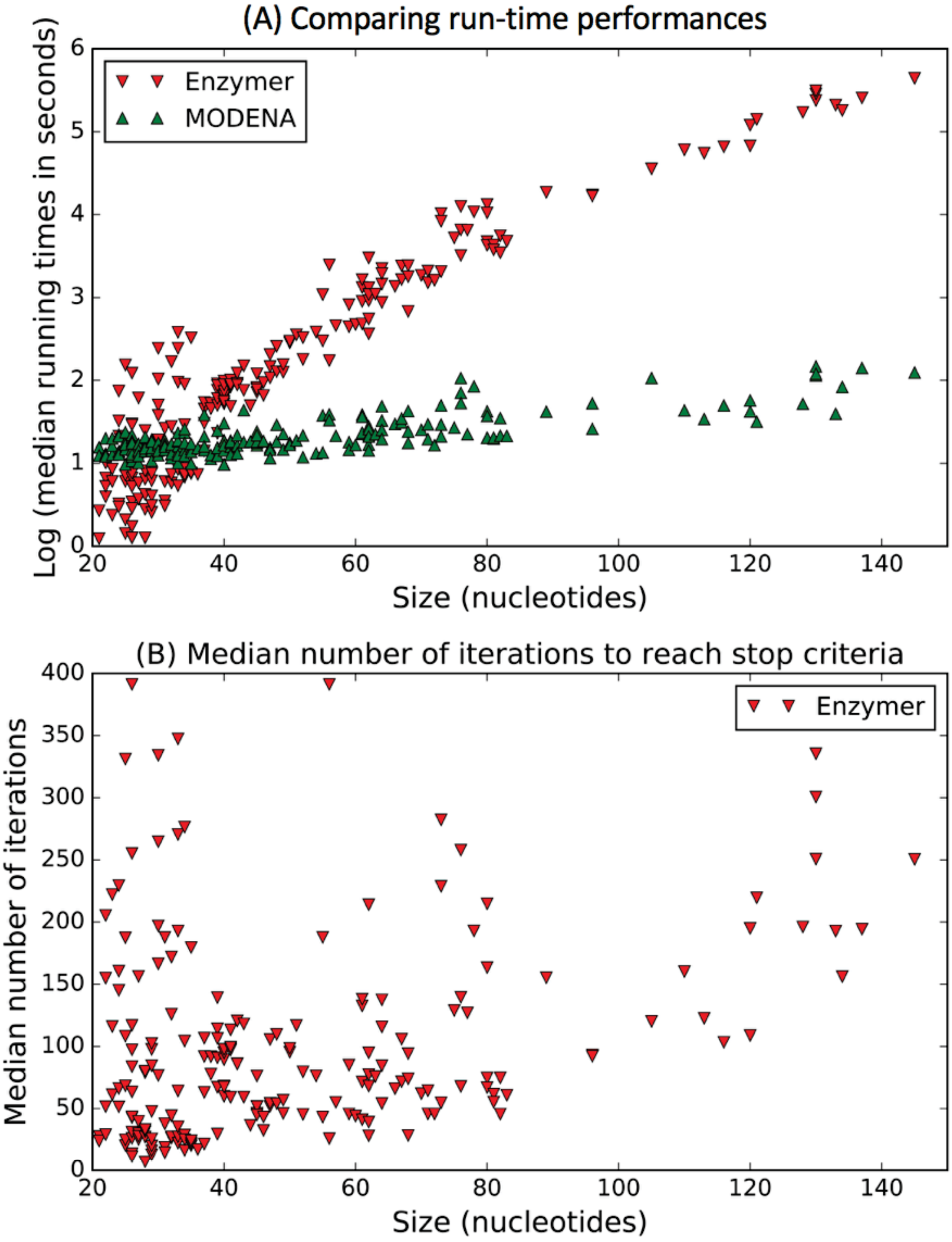


Figure 23: Run-time performance of design process - (A) Comparing run-time performance of `Enzymer` and `MODENA`. (B) `Enzymer` reached the stop condition in less than 200 iterations for 179 out of 201 cases.

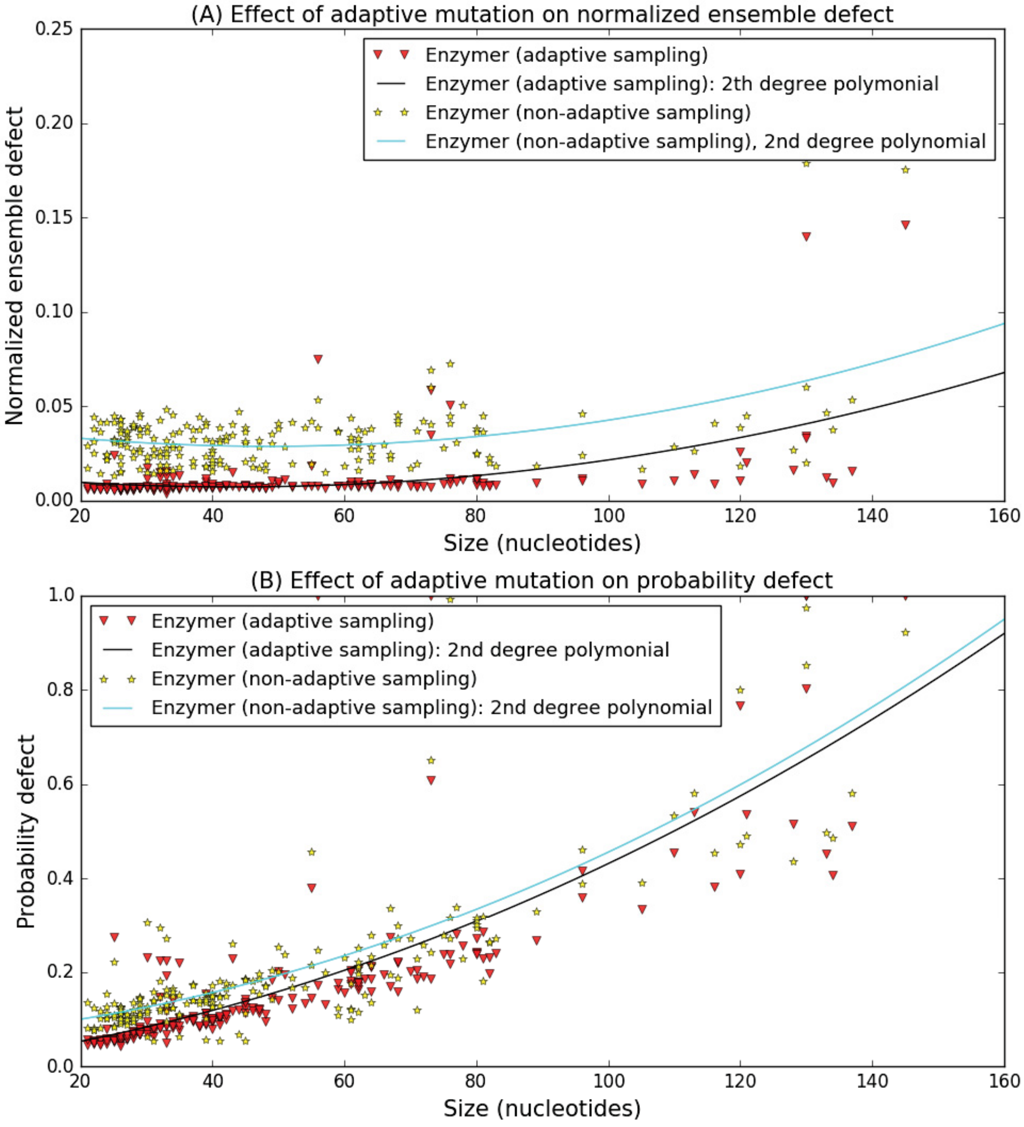


Figure 24: Effect of adaptive mutation - The adaptive sampling strategy lowered the median normalized ensemble defect in 199 cases (A) and also lowered the median probability defect of the sequences in 181 cases (B). Binomial test statistic with 99% confidence interval indicates for improving impact of the adaptive sampling strategy on both normalized ensemble defect and probability defect of the sequences we generated by `Enzymer`. For both figures the data was generated by setting $max\_it = 400$.

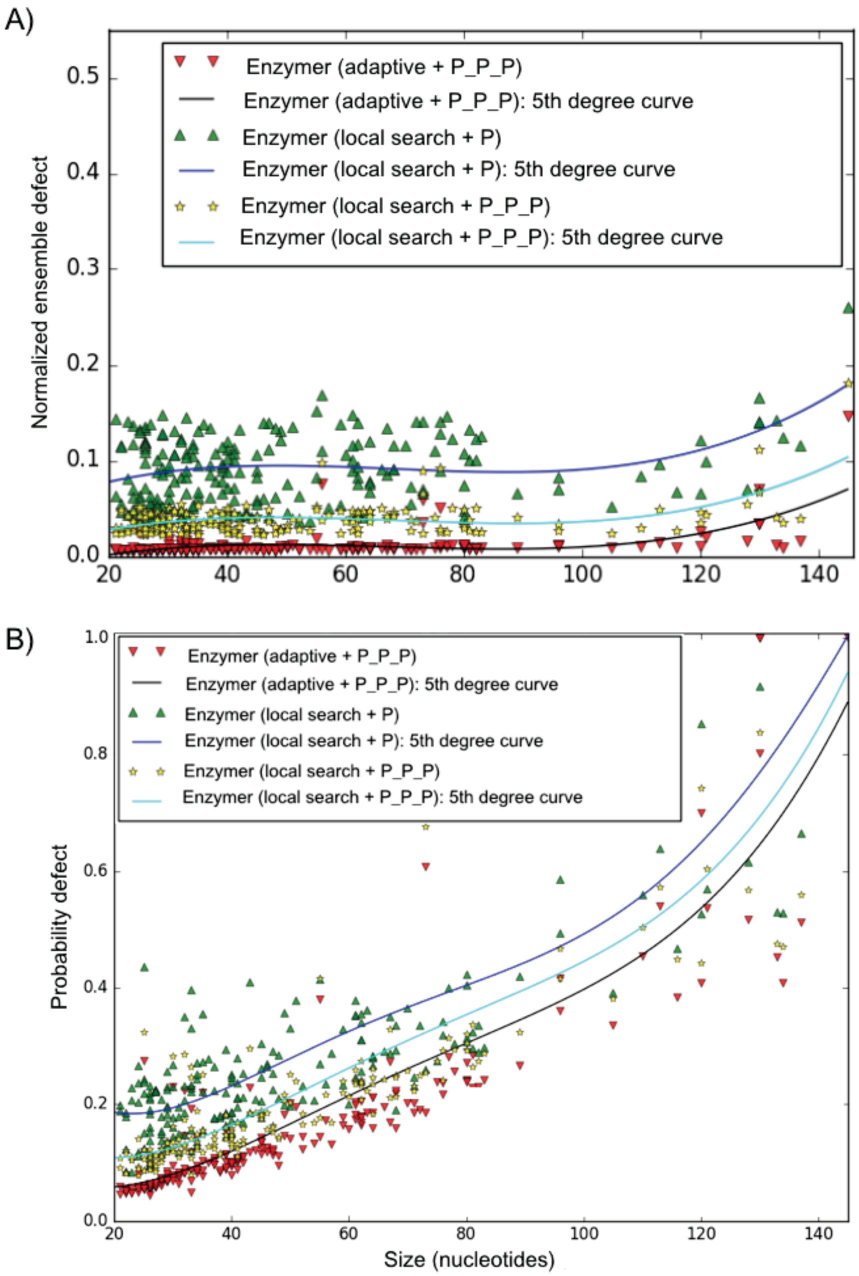


Figure 25: Effect of base pair distinction - Showing the effect of making distinctions between different types of base pairs (nested v.s non-nested) on normalized ensemble defect (A) and on probability defect (B) when a paired position is subject to mutation. In this figure **P** means at the time of each mutation, the type of the base pair (nested v.s non-nested) was not taken into account and the probability of mutation was only read from matrix $P$. On the other hand, **P_P_P** means at the time of mutation of a paired position, a distinction between nested and non-nested pairs were was taken into account; hence the probability of the mutation was read from either of $P'$ or $P''$ respectively.

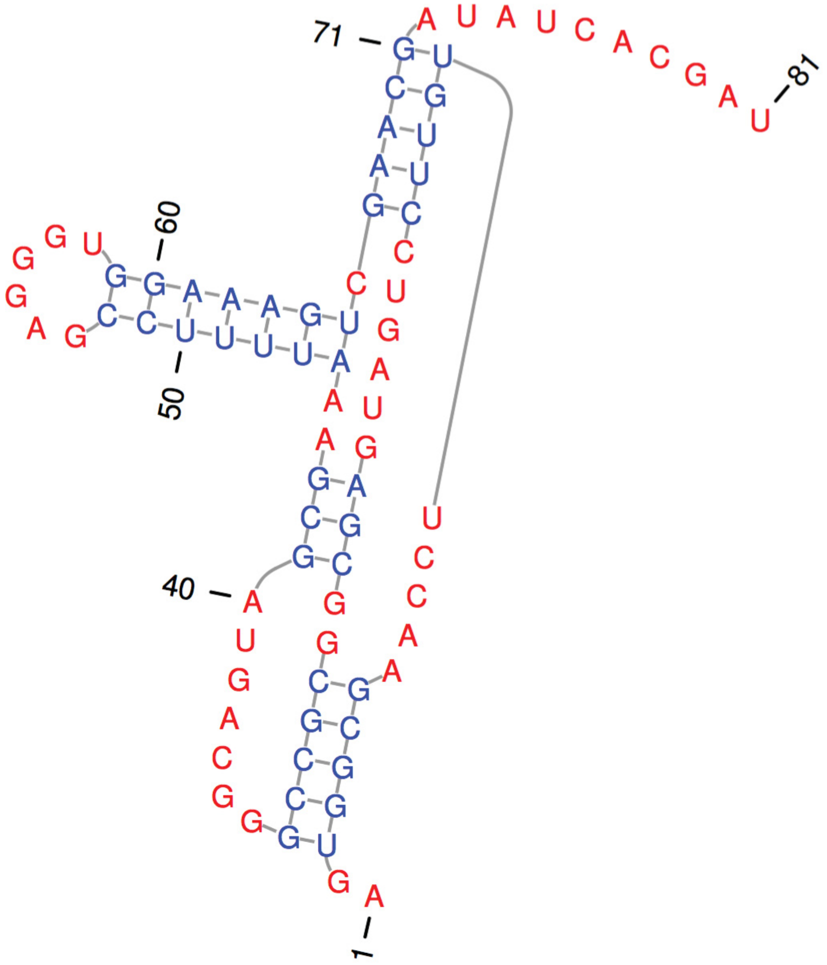


Figure 26: Secondary structure of hammerhead ribozyme from mouse gut metagenome - The stems are in blue and free nucleotides are in red. The five nucleotide long pseudoknot, starts at position 3 on stem 1. The sequence represents the $\phi^1_{HH}$ sequence designed by Enzymer. The secondary structure in standard dot bracket notation is presented by "..[[[[[.....(((((......(((..]]]]].......)))..(((((((......))))))).)))))..........” and is extracted from (Perreault et al., 2011). We used PseudoViewer3 (Byun and Han, 2009) to generate this figure.

## 5.7 Discussion

### 5.7.1 Summary of contributions

We presented Enzymer, a complete RNA inverse folding pipeline to reengineer functional ncRNA secondary structures including pseudoknots. Enzymer uses the multiple sequence alignment data which represent conserved evolutionary signals and implements a novel adaptive defect weighted sampling algorithm for the design of pseudoknotted RNA secondary structures. Enzymer (1) uses multiple sequence alignment data to generate design templates for functional RNAs, (2) uses NUPACK to compute the equilibrium characteristics of RNA design candidates, (3) dynamically adapts the total number of positional mutations at each iteration during the run-time, and (4) chooses target positions for mutation in respect to their type (free nucleotide, nested base pair or non-nested pair) as well as their positional contribution to ensemble defect of the design candidate. To benchmark Enzymer, we used a biological dataset of naturally occurring pseudoknotted secondary structures from the PseudoBase library and compared our results with 1) MODENA which implements the NSGA II optimization algorithm and 2) antaRNA which implements an ACO algorithm.

### 5.7.2 Summary of results

**Quality of solutions:** Our benchmark dataset contains 201 naturally occurring pseudoknotted secondary structures of size 21-144 nucleotides. For each structure, we used Enzymer and generated 30 RNA sequences and compared our results with the results generated by MODENA and antaRNA. We showed that Enzymer explores the mutational landscape of the candidate RNAs more efficiently and generates sequences that have lower ensemble defect, lower probability defect and higher Boltzmann frequency than those generated by MODENA and antaRNA. We also showed the sequences designed by our method have similar thermostability when compared to the sequences generated by MODENA but show better thermostability when compared the sequences generated by antaRNA. Furthermore, we showed our method succeeds more often than both MODENA and antaRNA do.

We emphasize that Enzymer extends the NUPACK design algorithm so that it can also include pseudoknots. However, if no pseudoknot is present in the target structure, our method will simply call the original NUPACK-design algorithm to generate sequences for pseudoknot-free targets.

**Run-time performance:** We observed that in 89% of the cases where the size of the target structure is bellow 144 nucleotides, our method can generate sequences with normalized ensemble defect value bellow 0.01 in less than 200 iterations. We also demonstrated that our adaptive sampling strategy causes the algorithm to reach the stop criteria in fewer iterations and therefore reduces the computational cost associated with the sampling process (Figure 24). Given our simulation results in respect to the run-time requirement of our approach, we conclude that our method is an excellent choice for the design of pseudoknotted RNA secondary structures of size up to 150 nucleotides. To our knowledge, there exists no other pseudoknotted RNA secondary structure designer algorithm that generates sequences that match the quality characteristics of sequences generated by `Enzymer`. Further experimentation will allow one to obtain a more accurate understanding about the applicability of `Enzymer` on larger and more diverse structures.

### 5.7.3 Novelty and significance

**Novelty of our optimization strategy in context of RNA design:** `Enzymer` is the first computational design method which makes use of the Dirks energy model to design functional pseudoknots. This is important because the applicability of an energy model first must be verified *in-silico* before moving to wet-lab experimentation. Also, in the context of RNA design, this is the first method which makes a distinction between the nested and non-nested base-pairs in a given secondary structure. As we see in Figure 25, making distinction between the different types of pairs leads to faster a convergence towards sequences with higher qualitative values. The mix of usage of a new energy model, as well as the *adaptive sampling* search which makes a distinction between the key structural attributes makes `Enzymer` a novel RNA designer method.

**Significance of our results in context of RNA design:** Our algorithm reaches the highest quality for pseudoknots compared to any other method reported in the literature. This accomplishment is significant as there are studies such as (Dotu et al., 2014) which demonstrate how a small change in quality introduces significant negative effects in functionality of the designed molecules when tested in biological settings. On another point, we demonstrated that the Dirks energy model can result in high quality functional pseudoknots; this is particularly important since the Dirks energy model was not initially designed for this purpose. Our results suggest that any arbitrary energy model may be useful for a more diverse range of modelling and design contexts than initially thought.

**Novelty of our method as a generic optimization strategy:** Our adaptive weighted sampling algorithm dynamically chooses the distance between the current solution and the next candidate solution. When the quality of the solution drops during the search and optimization process, our method takes the liberty to look inside the relatively far sides of the mutational landscape from the current candidate solution. Conversely, when the quality of the candidate solutions converge to higher quality values, our method looks in relatively smaller proximities of the current solution. As we see in Figure 24, such an adaptive search method contributes positively to the ability of the optimization method enabling it to converge faster to higher quality solutions. While there are reports in the literature where some form of memory (Tang and Miller-Hooks, 2005) or other heuristics (Hansen and Mladenović, 2014) are used to adaptively guide the search method, none have used Boltzmann sampling as a way to dynamically decide on the distance of current solution and the next candidate solution. To the best of our knowledge our approach in determining how far a candidate solution from iteration $n$ and iteration $n + 1$ positioned from one another is new.

**Significance of our results in the context of combinatorial optimization:** To characterize and understand the performance of our optimization method as a generic search and optimization algorithm, we decoupled the fitness evaluation from the search strategy and presented the results in Figure 22. We observed that our adaptive weighted sampling method can reach higher quality solutions while requiring a smaller number of fitness evaluations as compared to two of the most successful combinatorial optimization methods (NSGA II and ACO). This is particularly significant when the fitness evaluation is a costly task. In a recent work (Taneda, 2015), NSGA II is coupled with an ensemble defect optimization objective in order to design pseudoknotted RNAs with low defect. However, the author concluded that designing molecules larger than 80 nucleotides is not feasible. The incapability of NSGA II to scale as well as our method does, is another testimony of the significance of our method as a generic optimization strategy. This observation is even more significant when we take into consideration how evolutionary approaches —when used in a combinatorial optimization setup —often show superior performance compared to other optimization strategies such as various flavours of local search.

### 5.7.4 Constrained sequence design to reengineer a Hammerhead ribozyme

Our complete design pipeline allows one to insert evolutionarily conserved motif sequences extracted from multiple sequence alignment data to generate design templates that are useful to reengineering naturally occurring RNAs with known function. We used a naturally occurring Hammerhead motif and `Enzymer` to reengineer a *cis*-acting Hammerhead ribozyme from the mouse gut metagenome. Our method achieved mean and median normalized ensemble defect values of 0.046 and 0.047, respectively. Future experimentation will allow us to better understand the applicability of our design pipeline as well as the applicability of the particular energy model we used to re-engineer *cis*-acting Hammerhead ribozymes.

### 5.7.5 Limitations

We note that the applicability of `Enzymer` is bound by the ability of `NUPACK` to recognize different classes of pseudoknots. `NUPACK` realizes pseudoknots for single RNA strands such that the search space can be broken into all secondary structures that can be decomposed into two pseudoknot-free structures. Due to this limitation, when we used `NUPACK` to filter the original dataset, which was provided by (Taneda, 2012), the number of structures were reduced from 266 to 201. However, to our knowledge `NUPACK` is the only available computational framework, which can compute the partition function for a limited but biologically relevant class of pseudoknots. Hence, `NUPACK` is the best choice of the folding algorithm to design pseudoknotted RNAs with low ensemble defect, low probability defect and high thermostability.

### 5.7.6 Future work

To our knowledge neither `Enzymer` nor any other existing sequence designer algorithm exists which can design RNA sequences for multi-strand and multi-target models such as the *trans*-acting glmS ribozyme described by (Klein and Ferré-D'Amaré, 2006) or the oligonucleotide-sensing allosteric ribozyme based logic gates such as the ones described by (Penchovsky and Breaker, 2005) if pseudoknots are present.

One can use `NUPACK` to compute the equilibrium characteristics of pseudoknot-free complexes of interacting RNA species (Wolfe and Pierce, 2014), or use `NanoFolder` (Bindewald et al., 2011) to predict base pairings of pseudoknotted complexes of interacting RNA species. As a future work, we intent to use `NUPACK` and `NanoFolder` as folding algorithms to build on our adaptive defect weighted sampling algorithm in order to include the ability to design RNA sequences for multi-strand and multi-target secondary structures where pseudoknots can be present in single stranded forms. Such an improvement will open a door to design oligonucleotide sensing genetic networks that implement more complex modular interactions such as networks of interacting RNA species where each single stranded RNA species can include pseudoknots.

### 5.7.7 Acknowledgments

Computations were made on the supercomputer Guillimin at McGill University, managed by Calcul Quebec and Compute Canada. The operation of this supercomputer is funded by the Canada Foundation for Innovation (CFI), Ministere de l'Economie, de l'Innovation et des Exportations du Quebec (MEIE), RMGA and the Fonds de recherche du Quebec —Nature et technologies (FRQ—NT).

### 5.7.8 Author contributions

Kasra Zandi (KZ) designed the study, proposed the methodology, implemented the software, generated results, conducted the analysis and wrote the manuscript in its entire form. KZ revised the manuscript to address the issues raised by the reviewers. Gregory Butler (GB) provided oversight to the research process, provided comments and corrective remarks regarding the methodology and the analysis. Nawwaf Kharma (NK) provided supervision for the research process related to this article, monitored the discussion sessions, read the manuscript and provided corrective remarks about the methodology, implementation and analysis. Authors declare no conflict of interest.

# Chapter 6

# Computational design and experimental validation of pseudoknotted ribozymes

## 6.1 Preface

In Chapter 5 we introduced a novel RNA design methodology named Enzymer for efficient design of RNA secondary structures including pseudoknots. In this chapter we provide biological evidence that the RNAs designed by Enzymer are functional in biological contexts. The content of this chapter except for section 6.4.4 and related paragraphs has been submitted to the RNA J2ournal (impact factor 4.94) (Zandi et al., 2018).

## 6.2 Abstract

The design of new RNA sequences that retain the function of a model RNA structure is a challenge in bioinformatics because of the structural complexity of these molecules. RNA can fold into secondary and tertiary structures by forming stem loops and pseudoknots. Pseudoknots are secondary structure motifs where the loop of a stem base pairs with nucleotides of another stem or of a junction and this motif is very important for numerous functional structures. It is important for any computational design algorithm to take into account these interactions to give a reliable result. In our study, we validated synthetic ribozymes designed by Enzymer which implements algorithms allowing for the design of pseudoknots. Ribozymes are catalytic RNAs that have activities similar to those of enzymes. Ribozymes like the Hammerhead and the glmS have a self-cleaving activity that allows them to liberate the new RNA genome copy during circular replication or to control the expression of the *upstream genes*, respectively. We demonstrated the efficiency of Enzymer by showing that the pseudoknotted hammerhead and glmS ribozymes it designed were

active *in-vitro*. Finally we propose a novel gene regulatory architecture enabling a *cis-acting* hammerhead ribozyme to down regulate the expression of a reporter red fluorescent protein (RFP) gene in the presence of an external stimuli IPTG. We tested the proposed architecture *in-vivo*. Despite being inconclusive our *in-vivo* results suggest combining ribozyme sequences with protein coding sequences could potentially open the door to the successful design of novel gene regulatory networks with new functions.

## 6.3 Introduction

Non-coding RNAs (ncRNAs) play key roles in some key cellular processes. Among the most studied ncRNAs, we find micro RNAs (miRNAs), which are about 22 nucleotides (nt) in length and act as post-transcriptional gene silencing mediators (Huntzinger and Izaurralde, 2011). On the other hand, some long non-coding RNAs (lncRNAs) act as modular scaffold for histone modification (Tsai et al., 2010), in cell differentiation and development (Fatica and Bozzoni, 2014). Natural functions of ncRNAs are numerous, but ncRNAs also have various applications in engineering biological systems (Ausländer et al., 2010; Cameron et al., 2014; Liang et al., 2011; Prommana et al., 2013; Zalatan et al., 2015), therapeutics (Esposito et al., 2014; Lienert et al., 2014; Ruder et al., 2011), and in nano-technology (Afonin et al., 2013; Grabow and Jaeger, 2014; Shu et al., 2015). Some ncRNAs such as riboswitches with more complex structures, act as receptors, binding specific *metabolite* and control gene expression (Serganov and Nudler, 2013). Other intricate RNA structures such as ribozymes can confer catalytic activities to RNA. The best known example is the hammerhead ribozyme (*HHRz*), which is involved in producing single-copy genomes out of the multimeric RNA resulting from rolling circle replication of viroids (Prody et al., 1986). There are also hundreds of examples of different hammerhead-type ribozymes for which RNA self-cleavage still has no obvious function (Hammann et al., 2012). In contrast, the glmS ribozyme has a clear regulatory function. This ribozyme cleaves the 5′-UTR (UnTranslated Region) of the mRNA where it is found by using glucosamine-6-phosphate (GlcN6P) as a *cofactor*, leading to degradation of the mRNA and repression of genes involved in GlcN6P synthesis when the latter is in sufficient concentrations (Collins et al., 2007; Winkler et al., 2004).

It is widely accepted that the function of ncRNAs is attributed to their structure (Leontis et al., 2006; Mortimer et al., 2014), which is determined by the nucleotide composition of the RNA polymer. An RNA strand folds to form an ensemble of secondary structures which then form stable tertiary structures. The diverse range of functions of ncRNAs, as well as the relationship between their sequence-structure and function, highlight the importance of methods for analysis and design of ncRNAs with desired structural attributes.

Formation of secondary structure is the first step in RNA folding. It starts by forming hydrogen bonds between the bases. For an RNA sequence $\phi$ with length $n$, a secondary structure is defined by a set of base-pairs $(\phi_i, \phi_j)$ where $1 < i < j < n$ such that positions $i$ and $j$ are paired.

For two base-pairs $(\phi_i, \phi_j)$ and $(\phi_k, \phi_l)$ a non-nested loop or a pseudoknot forms if either of the nesting rules $i \leq k \leq j \leq l$ or $k \leq i \leq j \leq l$ is violated. Pseudoknots are abundant in nature, exist in specific types (Condon et al., 2004) and are known to play key roles in the functionality of active RNAs (Staple and Butcher, 2005), including rRNAs (Powers and Noller, 1991), riboswitches (Gilbert et al., 2008) and ribozymes (Harris et al., 2015). Experimental methods, such as Nuclear magnetic resonance (NMR) spectroscopy (Varani and Tinoco, 1991) or X-Ray crystallography (Muchmore et al., 1996) used for determining RNA structure, are complex and time-consuming. Hence, computational approaches provide an attractive alternative to study and analyze RNA structures. In addition, computational methods can generate thousands of RNA sequences in short amounts of time, at relatively low cost, while providing the means to predict key sequence-structure attributes useful for further analysis and experimentation.

The classical definition of computational RNA structure prediction or RNA folding (Nussinov and Jacobson, 1980) is to find the set of base-pairs which are predicted to exist in RNA's most stable structure called the minimum free energy (MFE) structure at thermodynamic equilibrium. The MFE structure can be predicted by computational methods using some energy model. Over the past few decades, several energy models (Dirks and Pierce, 2003; Freier et al., 1986; Mathews et al., 1999; Serra and Turner, 1995) as well as computational structure prediction methods such as RNAfold (Hofacker, 2003), mfold (Zuker and Stiegler, 1981), HotKnots (Ren et al., 2005), RNAstructure (Reuter and Mathews, 2010), IPknot (Sato et al., 2011), pKiss (Theis et al., 2010), NUPACK-analyze (Zadeh et al., 2011a) have been developed. On the other hand, the RNA inverse folding problem is to find a sequence with an MFE structure that precisely matches a desired target structure. Besides the classical MFE as the design criterion, other criteria such as Boltzmann probability or ensemble defect optimization criteria have also been used. In the context of RNA inverse folding, Boltzmann probability quantifies the probability of an RNA sequence folding into a given structure, where the ensemble defect quantifies the expected number of incorrectly paired nucleotides at thermodynamic equilibrium (Dirks and Pierce, 2003). It has been shown that the sequences designed to maximize the Boltzmann probability or minimize ensemble defect tend to be thermodynamically more stable than those designed to satisfy the MFE criteria (Zadeh et al., 2011b; Zandi et al., 2016).

The RNA design problem is computationally difficult (Schnall-Levin et al., 2008) and to find solutions, most existing algorithms resort to heuristics and a combination of local, global and stochastic search methods. Generally, a random seed RNA sequence is first generated, then the seed is iteratively mutated until the predicted folding attributes of the design candidate converg to the desired values. Our recent work, Enzymer (Zandi et al., 2016) utilizes an adaptive weighted sampling strategy to design RNA secondary structure with low ensemble defect. The *in-silico* simulations showed that Enzymer generates RNAs that are thermodynamically more stable, have higher Boltzmann probability of folding into the desired target and have lower ensemble defect than those generated by other state of the art pseudoknot designer methods such as MODENA

(Taneda, 2011) and antaRNA (Kleinkauf et al., 2015).

The wealth of existing computational RNA secondary structure design methods, where each method utilizes a different design criterion and sequence optimization strategy, makes it a difficult task for experimental biologists to choose the right method for their particular RNA design purpose. For instance, some methods such as NanoTiler or NUPACK-design are more suited for the design of non-functional RNA nano structures, while other methods such as Frnakenstein are specialized in designing RNA switches. Another decisive factor in the applicability of an RNA designer method is its ability to realize of all the structural elements necessary for a particular design objective. For instance, only NanoTiler (Bindewald et al., 2008) MODENA, antaRNA and Enzymer can handle pseudoknots. Another important consideration is related to the applicability of the underlying energy model used by each design method. Still another important consideration is the availability of experimental evidence in support of the applicability of a design method for a specific design purpose. Ultimately, it is wet-lab experimental data that provide the most reliable measures of the usefulness of a design method. To the best of our knowledge, as summarized by (Churkin et al., 2017) there seems to be no comprehensive report in the literature that provides experimental evidence on the applicability of all inverse RNA folding methods in the design of pseudoknotted ncRNAs.

In this study, we demonstrate that Enzymer can be used as a reliable method for the design of pseudoknotted ribozymes. We used Enzymer to reengineer three naturally occurring ribozymes: a self-cleaving HHRz from the mouse gut metagenome, a self-cleaving HHRz from *Yarrowia lipolytica* (Perreault et al., 2011) as well as a self-cleaving glmS ribozyme (Klein and Ferré-D'Amaré, 2006). For each ribozyme, we obtained the minimal required catalytic core extracted from multiple sequence alignment data and used the catalytic core as a design template for Enzymer. We generated a population of candidate sequences for each ribozyme. For each ribozyme, we sorted the generated sequences by their predicted normalized ensemble defect value, chose a small set for *in-vitro* studies and measured their catalytic activities. We designed pseudoknotted HHRzs active to levels comparable to the wild type sequences, and we also designed a non-HHRz: the GlcN6P-dependent glmS ribozyme. Finally we describe a new architecture for combining the sequence of HHRz of *Yarrowia lipolytica* with the coding sequence of RFP and study the effect of HHRz on expression of RFP *in-vivo.*

## 6.4 Results

### 6.4.1 Synthetic hammerhead ribozymes with a pseudoknot for the stem I-II interaction

The activity of an RNA depends on its folding in defined secondary and tertiary structures (Bhartiya and Scaria, 2016). To have a functional synthetic RNA the structure of the original RNA used as template should be conserved. In our study, we used three different ribozymes as templates to design sequences with Enzymer that fit these functional structures and tested them. To validate the efficiency of Enzymer's designed sequences we tested experimentally two types of self-cleaving ribozymes; HHRz and GlmS ribozyme. For the hammerhead, two ribozyme structures from different genomes and contexts were used as templates: one from the mouse gut metagenome and the other from *Yarrowia lipolytica* (Perreault et al., 2011). Both of them are type I hammerhead ribozymes having a pseudoknot formed between stem I and the loop of stem II.

The HHRz from the mouse gut metagenome was chosen because its stem II is only two base-pairs (Figure 27 A, B), thus making it a good model to test Enzymer since misfolding could easily prevent stem II from folding correctly following a faulty design. Using the secondary structure of this HHRz (Figure 27 A,B), Enzymer generated 14 sequences to test and compare their activity with the wild type ribozyme. The sequence MM-HHRz3 could not be transcribed properly, but the other thirteen ribozymes were active compared to the wild-type which cleaved at 84% (Figure 27 1C). Ribozymes 4, 5, 11, 13 and 14 had higher cleavage activities than the wild type with cleavage respectively 96%, 88%, 87%, 92% and 88%. The rest of novel ribozymes showed variable cleavage efficiency either similar to the wild type (7, 8, 9, and 12) or less active (1, 2, 6, and 10) (Figure 27 C,D). By these results we demonstrated that Enzymer was able to design active hammerhead ribozymes using the inverse folding approach by conserving the pseudoknotted structure.

We initially planned to assay ribozymes at different temperatures, so the parameters used to design HHRz 1, 2, 4, 5 and 6 were set for optimal folding at 37°C, HHRz 13 and 14 at 17°C, 11 and 12 at 27°C, 7 and 8 at 47°C and 9 and 10 at 57°C. However, all ribozymes were assayed for self-cleavage and therefore cleaved during transcription at 37°C. While there might be a weak trend of better activity for HHRzs designed for lower temperatures, there is no significant difference between them.

### 6.4.2 Pseudoknotted hammerhead ribozymes overlapping coding sequence

Using Enzymer, we designed ribozymes with a sequence flanking the beginning of the coding sequence for the RFP. As a template, we used the *Yarrowia lipolytica* hammerhead (Figure 28 A, B). Sequence constraints include the conserved catalytic core, the *Shine-Dalgarno* sequence and the first 10 codons of RFP (Figure 28 A). The identity of the rest of the nucleotides could vary in order to accommodate these constraints and permit proper folding of the HHRz, while ensuring that we did not modify to the RFP protein. The aim is to have a non-defective protein and to be sure that if there is a decrease in protein expression it is caused by ribozyme's activity and not by a mutation in the protein sequence.

To have uncleaved size markers, we modified both ribozymes in the catalytic core without affecting the coding sequence. The ribozymes were tested for their cleavage efficiency during *in vitro* transcription and their activity was compared to the wild type. Both ribozymes were active with cleavage efficiencies of 76% for the YLHHRz_1 and 83% for the YLHHRz_2 not far from the

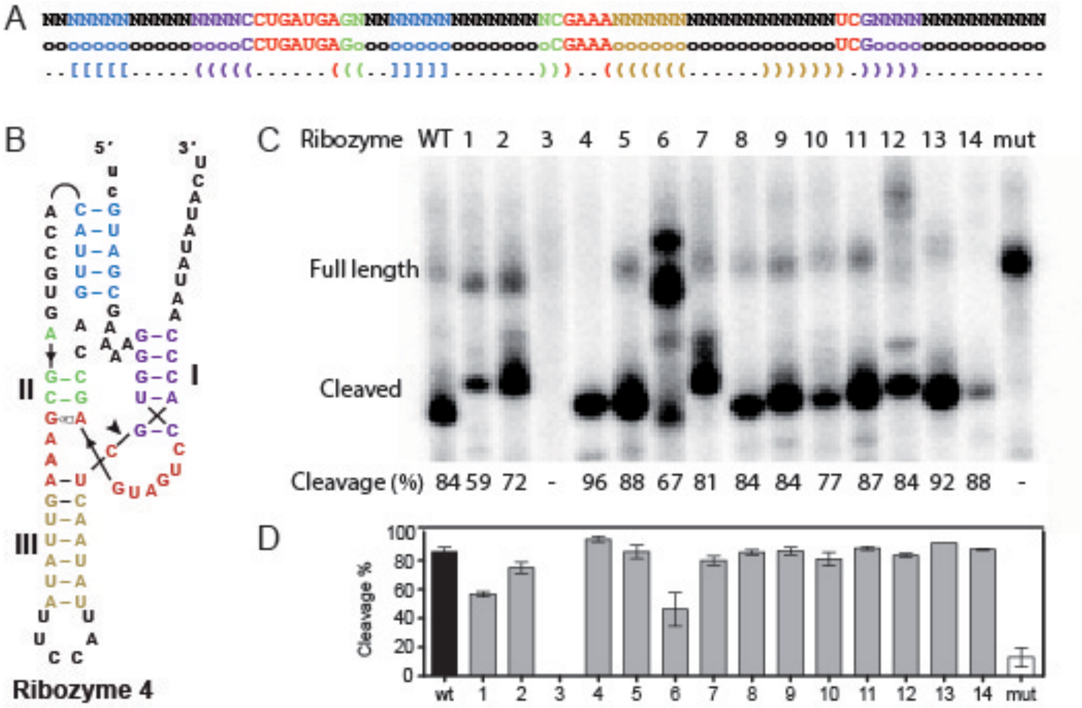


Figure 27: (A) Design template and secondary structure of the mouse gut metagenome HHRz structure used as the input for Enzymer. All of the original sequence was modified except the red nucleotides forming the catalytic core. (B) Secondary structure with a new sequence generated by Enzymer, with a colour scheme corresponding to that of the design template in A. Cleavage site is shown by an arrow head. (C) Cleavage activity of the ribozymes during transcription compared to the WT. (D) Cleavage activity of the ribozymes. Average of three experiments with standard deviation shown.

cleavage efficiency of the wild type (86%) (Figure 28). In both the active and inactive versions of the constructs, the expression is lower than in the no-ribozyme controls, suggesting that the HHRz stems hinder translation, whether cleavage occurs or not.

### 6.4.3 Synthetic glmS ribozymes activated by glucosamine-6-phosphate

To be able to see if Enzymer could successfully design structures more complex than the HHRz, we used the glmS ribozyme as another template. This ribozyme has a multi-stem structure, three pseudoknots and many highly conserved nucleotides (Figure 29 A,B). To design such a ribozyme we used the glmS ribozyme from *Thermohermoanaerobacter tengcongensis* (Klein and Ferré-D'Amaré, 2006).

We used two glmS sequences generated by Enzymer to compare their activity to the wild type glmS ribozyme. These ribozymes were transcribed in the presence or absence of GlcN6P. We also modified the catalytic core of the wild type glmS ribozyme to use as an uncleaved size marker and an RNA corresponding to the cleaved part of the ribozyme was used as a cleaved size marker. The wild-type ribozyme has a cleavage endpoint of 76% and glmS_2 designed by Enzymer had 45%

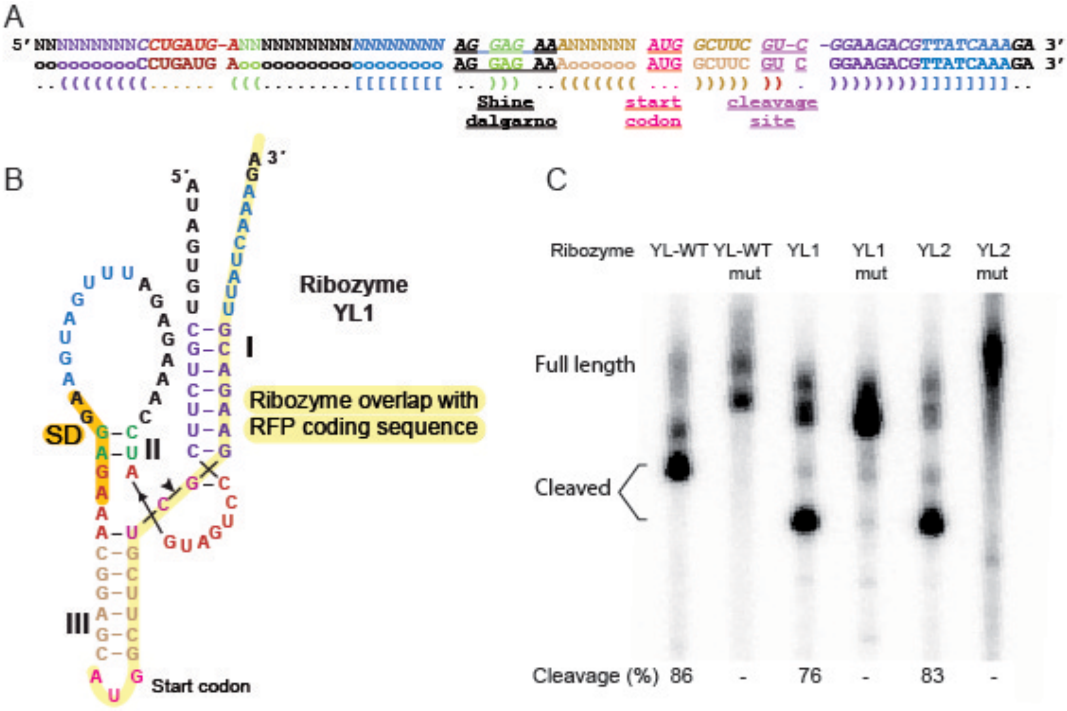


Figure 28: (A) Design template and secondary structure of the *Yarrowia lipolytica* HHRz structure used as the input for Enzymer. The original sequence of the ribozyme was modified except for the red nucleotides forming the catalytic core. Nucleotides corresponding to sequence constraints for fusion with the coding sequence and the Shine-Dalgarno are indicated. (B) Secondary structure with a new sequence generated by Enzymer, with a color scheme corresponding to that of the design template in A. Portions corresponding to the sequence of the RFP gene are underlined in yellow for the coding sequence and in orange for the Shine-Dalgarno (SD). The cleavage site is shown by an arrow head. (C) The cleavage activity of the ribozymes during transcription as compared to the WT ribozyme. Cleaved products of YLHHRz_1 and YLHHRz_2 are different from wild type because we reduced stem III by four base pairs and one nucleotide from the loop as compared to WT to permit proper spacing between SD and start codon.

cleavage efficiency while glmS_1 was inactive (Figure 29 C). As with the wild-type, self-cleavage of glmS_2 is completely GlcN6P dependent due to no detectable cleavage, showing that in addition to self-cleaving capabilities, glmS_2 also has a competent binding pocket for GlcN6P.

### 6.4.4 Effect of hammerhead ribozyme on expression of RFP

The two YLHHRz_1 and YLHHRz_2 sequences which overlapped the coding sequence of a RFP were tested *in-vivo* by inserting each into a *plasmid* which was then transformed into BL21DE3 (Pan and Malcolm, 2000) bacteria. After cultivation of the bacteria overnight, the amount of fluorescence emitted from the RFP was measured for each bacterial culture and then normalized (Figure 30). Each construct and the corresponding mutant constructs were cultivated in three different bacterial cultures. In Figure 30 each group of bars corresponds to the level of fluorescence emitted in the presence of IPTG at three different levels. In the absence of IPTG, this system is constitutively repressed and is expected to show very low levels of RFP. By adding IPTG, we expect to observe a reduction in the expression of RFP as compared to the positive control. For the positive control (the RFP+ column) we observe high amounts of fluorescence and for the negative control column (bl21) we observe relatively low amounts of fluorescent. Note that there is no need to add IPTG to the positive and negative control cultures.

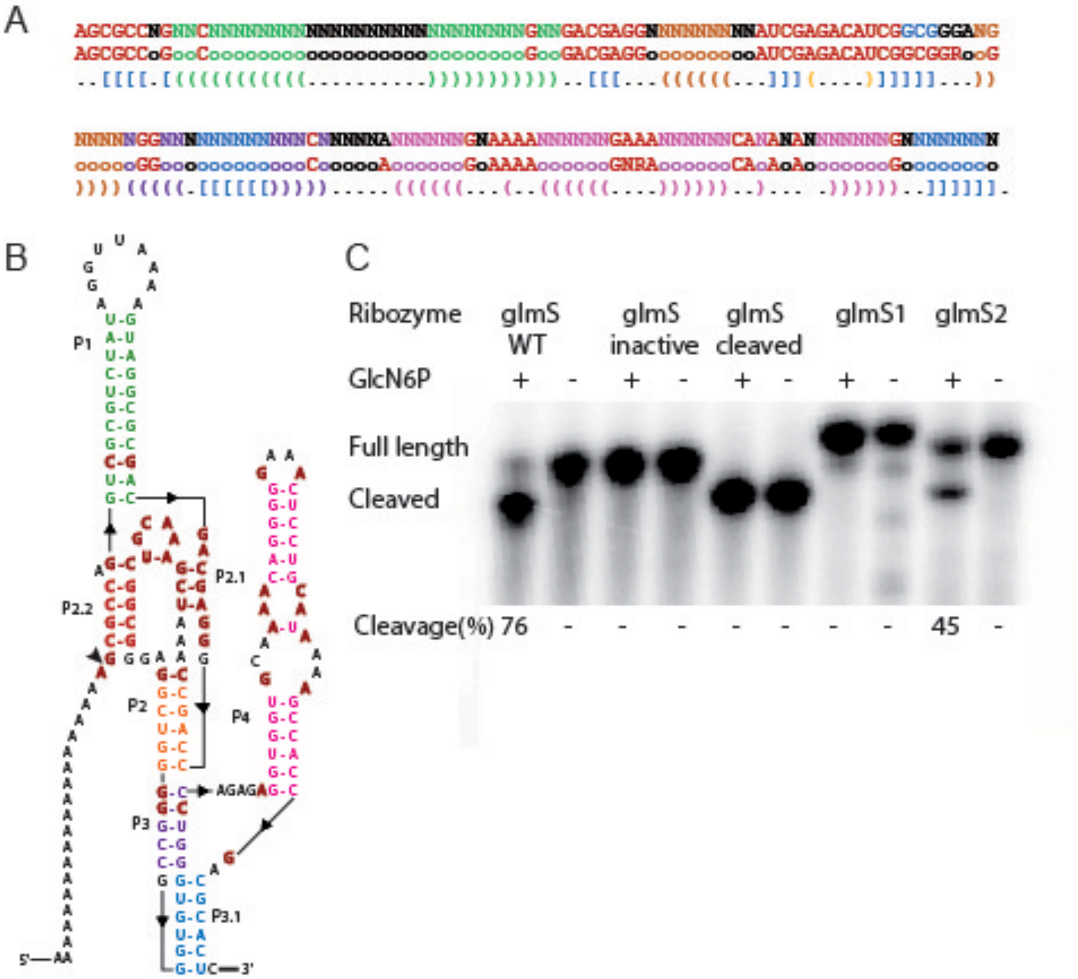


Figure 29: (A) Design template and secondary structure of the glmS ribozyme used as the input for Enzymer. All of the original sequence was modified except the red nucleotides forming the catalytic core. (B) Secondary structure with a new sequence generated by Enzymer, with a colour scheme corresponding to that of the design template in A. Cleavage site is shown by an arrow head. Red nucleotides in bold are those conserved from the WT sequence. (C) Cleavage activity of the ribozymes during transcription compared to the WT with and without glucosamine-6-P (GlcN6P). Inactivating mutations were made to the WT sequence to get the glmS inactive size marker and glmS cleaved is a shorter RNA corresponding to the expected size of the cleaved product.

In the bacterial population including the YL1 ribozyme, in two out of three cases (YL1(1) and YL1(2)) we observe a significant decrease in the amount of fluorescent when IPTG is added and for the case of YL1(3) we observe only a small reduction in the expression of RFP. We also observe for the bacterial population containing the mutant strains of YL1 in two out of three cases (YL1(2)m and YL1(3)m) the addition of IPTG did not lead to a significant change in the amount of fluorescence. We obtained similar results for YL2. Based on our results it appears the bacterial cultures expressing the designed ribozymes tend to emit lower the amounts of fluorescence when IPTG is added to the medium compared to the cultures expressing the mutant ribozymes. Overall in seven out of 12 cultures (YL1(1), YL1(2), YL1(2)m, YL1(3)m, YL2(2), YL2(3) and YL2(1)m) the measured relative light emission levels are aligned with our expectations. Our results are inconclusive however the data suggests that is may be possible that the reduction in expression of the RFP reporter gene is caused by the ribozyme activity.

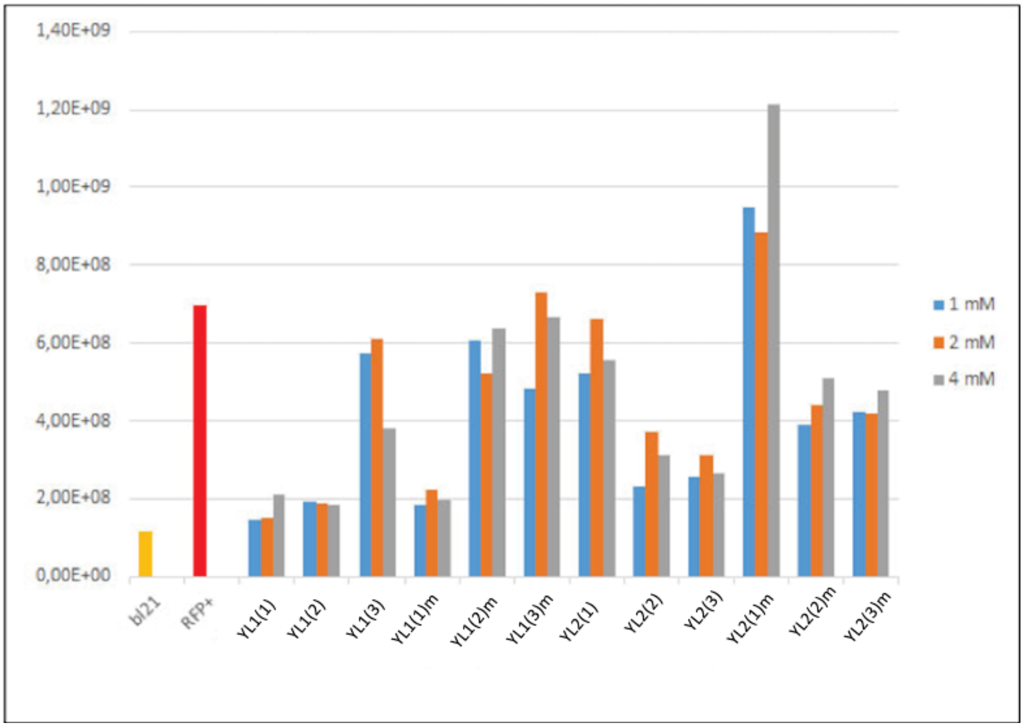


Figure 30: Each group of bars corresponds to the normalized level of fluorescence emitted using three different levels of IPTG.

## 6.5 Discussion

In this work we proved that Enzymer can successfully design functional pseudoknotted RNAs. We tested experimentally a total of 18 ribozymes designed by this new inverse folding algorithm. Despite the fact that glmS_1 was inactive, the rest of the results show that Enzymer successfully designed pseudoknotted functional ribozymes that were active *in-vitro*. We also demonstrate for the first time that the Dirks energy model (Dirks and Pierce, 2003), which was developed for characterization of pseudoknotted RNA shapes, can be effectively used to model and design functional ribozymes. Notably despite previous attempts to reengineer the glmS ribozyme (Lau and Ferré-D'Amaré, 2013, 2016; Lau et al., 2017) the glmS sequence we designed has the least sequence similarity with the wild-type sequence for any designed glmS ribozyme. Interestingly, both glmS_1 and glmS_2 diverge slightly from the more recent alignment consensus found (McCown et al., 2011) but glmS_1 diverges at 16 positions, vs 11 for glmS_1, that differs from positions with at least 75% of conservation. This suggests that rather than being a faulty Enzymer design, glmS_1 would likely be inactive because the provided model lacked important information. Indeed, glmS_1 was generated in our early attempts to design a sequence for this complex ribozyme and as such it lacked some constraints (underlines in Table 3). In other words, all of the ribozymes designed with an appropriate model were active. Despite being inconclusive our *in-vivo* results lead us to hypothesize that hammerhead ribozymes overlapped with coding sequence of a reporter gene could potentially facilitate modulation of the expression of the target gene. The validity of this hypothesis requires further investigation.

To our knowledge, only one group actually tested ribozymes designed through inverse folding (Dotu et al., 2014), where they used a type III HHRz from a portion of the plus polarity strand of Peach Latent Mosaic Viroid (PLMVd). To make their design, they used RNAiFold and they tested the generated ribozymes experimentally. These ribozymes were active, but RNAiFold does not take into account pseudoknots. On the other hand, others have designed pseudoknotted ribozymes, like the HDV ribozyme, but without experimental validation (Taneda, 2012). This is, to our knowledge, the first experimental validation of pseudoknotted ribozymes designed by inverse folding.

The ability of Enzymer to design sequences for complex active structures with many sequence constraints, such as for the *Yarrowia lipolytica* HHRz, paves the way for combinations of sequence elements. With proper design many useful arrangements can be made to engineer new regulatory elements by overlapping important sequences, such as coding sequence or splicing sites, with structures, such as ribozymes or riboswitches, to gain new functions.

## 6.6 Materials and Methods

We obtained the secondary structures as well as the minimum catalytic core for the HHRzs from (Perreault et al., 2011). For the glmS ribozyme, we used the secondary structure as well as the minimum catalytic core from the model presented by (Klein and Ferré-D'Amaré, 2006) but in its natural cis-acting form. Given a secondary structure and a set of nucleotides representing the minimal catalytic core, the step to initialize a design template is to generate a seed sequence using the catalytic core and then using the letter “o” for any position which is not part of the catalytic core, or highly conserved nucleotides, of the ribozymes.

Figure 31 shows the architecture of the construct we devised to test the effect of overlapping hammerhead sequence with RFP. In our proposed architecture, we added a *DNA promoter* upstream of the ribozyme sequence to control triggering the transcription of the sequence. In our case we used a DNA transcription promoter which is blocked by the *LacI* repressor binding to it. When an external stimuli which in our case is IPTG is added to the medium, the IPTG binds to the LacI repressor and disallows it to tightly bind to the DNA promoter; thus allowing the transcription process to be triggered. The last component of our architecture is a gene located downstream of the ribozyme sequence. We overlapped 24 nucleotides from the $3'$ ending of the ribozyme sequence with coding sequence of a RFP reporter gene. To express the downstream gene *in-vivo*, we also needed to include a Shine-Dalgarno sequence before the RFP coding sequence.

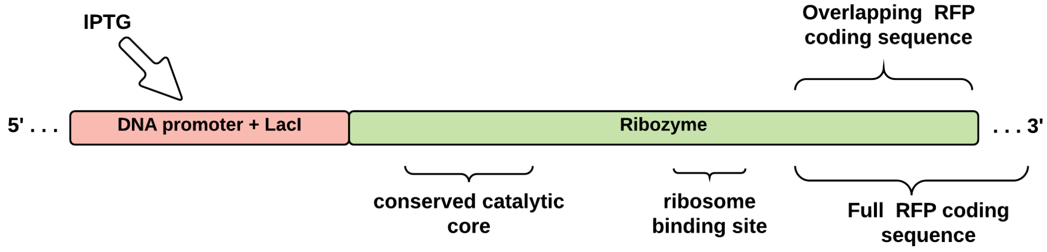


Figure 31: Architecture of a ribozyme system that can be triggered by addition of an external stimuli which in our case is IPTG. Without IPTG this system is repressed because the LacI repressor is blocking the transcription by binding to the DNA promoter. Addition of IPTG initiates transcription process.

We ran Enzymer using the design templates as input and set the maximum number of iterations to 600. We generated 14, 2 and 2 sequences for the mouse gut metagenome hammerhead, *Yarrowia lipolytica* hammerhead and the glmS ribozymes respectively in independent trials. For most trials, we chose the Mathews parameters (Mathews et al., 1999) but for optimal folding at temperatures other than 37°C we used the Serra and Turner energy parameters (1995). For all cases, we used the additional parameters for pseudoknots from the Dirks and Pierce model (2003). The sequences we generated for *in-vitro* analysis are presented in Table 3. The sequences generated for Figure 31 are presented in Table 4. In table 4 the coding sequence of the conserved catalytic core, RBS and RFP are in bold. In the mutant sequences, the nucleotides changed are underlined. The wet-lab protocols for the *in-vitro* experiments are presented in Appendices 2 and 3 respectively.

| **Label** | **Mouse gut metagenome hammerhead ribozyme RNA sequences** | **defect** |
|---|---|---|
| MHHRz_1 | UCGUAGCGAA AAGGGUCCUG AUGAGCCAGU UACACCGUAG GCGAAAGUUA UAUUCCAUUA UAACUCGACC CAAUAUAUAC U | $7.0E-02$ |
| MHHRz_2 | AACGGAGCCC UUCCGCCCUG AUGAGCAACU CUGAAUAAAA GCGAAACUGU AGAACUACCU ACGGUCGGCG GUUUUCUAUA C | $4.4E-02$ |
| MHHRz_3 | CUCUCCGAAA CUUGGUCCUG AUGAGGCCCG GAGUUAACCG CCGAAACUGC GUAAACCGAU GUAGUCGACC ACAAAAUACU A | $7.49E-02$ |
| MHHRz_4 | CCGUCGCAAA AAGGGUCCUG AUGAGCAAGC GACAAAAAAA GCGAAACCCG UCGAUUAAGA UGGGUCGACC CAAAAAAAAA A | $4.0E-02$ |
| MHHRz_5 | AAGUCCCAAA AAGGGCCCUG AUGAGCGAGG GACAAAAAAA GCGAAACUUC GUGAAAGAGC GAAGUCGGCC CGAGAAAAAA A | $7.6E-02$ |
| MHHRz_6 | GACGGCCCCC CCGGUCCCUG AUGAGCUAGG CCGAAAACGA GCGAAACCGA GAUCUUUUUC UCGGUCGGAC CUCUACCUAU U | $9.9E-02$ |
| MHHRz_7 | AAGCGGGAGA GAGGGGCCUG AUGAGUGACC CGCGAAAAAA ACGAAACCUG GUUCAGCUGC CAGGUCGCCC CAGGAGAAGU G | $9.5E-02$ |
| MHHRz_8 | AACCACCAAA AAGUGCCCUG AUGAGCGAGG UGGAAAAAAA GCGAAAGGGC CCAUGAACGG GUCCUCGGCA CAAAAAAAAA G | $9.1E-02$ |
| MHHRz_9 | AGCGGACAAG AAGCGGCCUG AUGAGUGAGU CCGGAAAAAA ACGAAAGGCA CUAGAUAGAG UGCCUCGCCG CAAAUAAAAC G | $1.0E-01$ |
| MHHRz_10 | AACCAGGAAA AAGGGCCCUG AUGAGCGACC UGGGAAAAAA GCGAAAGGUC CGAACGAGCG GACCUCGGCC CAAAAAAAAC G | $9.7E-02$ |
| MHHRz_11 | GAGAGACAAA AAAGGCCCUG AUGAGCGAGU CUCAAAAAAA GCGAAACGGG AUUGAUAUGU CCCGUCGGCC UGGAGGAAAG A | $8.2E-02$ |
| MHHRz_12 | AAGGUCCAAA AAGGGCCCUG AUGAGCACGG ACCAAAAAAC GCGAAACUCC AUGACAGAGU GGAGUCGGCC CACAAAAAAC C | $7.7E-02$ |

| | | |
|---|---|---|
| MHHRz_13 | AAGAGGGAAA AAGGUUCCUG AUGAGCGACC CUCAAAAAAA GCGAAACAGC ACGAGAAAGU GCUGUCGAAC CAAAAAAAGA A | $7.9E-02$ |
| MHHRz_14 | AAUGGCCAAA AAGGGUCCUG AUGAGCACGG CUAAAAAAAC GCGAAACCGG UCAGAUAAGG CCGGUCGACC CAAACAACAC C | $8.0E-02$ |
| MHHRz_3mut | AACUAGCAAA AAGCGGAGGG AUGAGGAAGC UAGAAAAAAA CCGACCGCAA GCAUAUACGC UUGCUCGCCG CAAAAAAACA C | $1.28E-01$ |
| Wild type | GGUACCGAAU AAAUCCCCUG AUGAGCAACG GUGAGAGCCG GCGAAACUAC CCAAACAAGG GUAGUCGGGA UAGUACCAUA A | $3.80-01$ |
| Design template | oooooooooo ooooooCCUG AUGAGooooo oooooooooo oCGAAAoooo oooooooooo ooooUCGooo oooooooooo o | − |
| Secondary structure | ..[[[[[... ..(((((... ...(((..]] ]]]....... )))..((((( ((......)) ))))).)))) )......... . | − |
| | *Yarrowia lipolytica* hammerhead ribozyme RNA sequences | |
| YLHHR_1 | AUAUACCCGU CUUCCCUGAU GAUCCAAAAA AAUUUGAUGA AGGAGAAACG AGGCAUGGCU UCGUCGGAAG ACGUUAUCAA AGA | $8.0-E02$ |
| YLHHRz_2 | AUAGUGUCGU CUUCCCUGAU GAUCCAAAGA GAUUUGAUGA AGGAGAAACG GAGCAUGGCU UCGUCGGAAG ACGUUAUCAA AGA | $8.0-E02$ |
| Wild type | GGGGGACUGG CUGCCCUGAU GAGAACAAAC CCAUGACUAG CGUCGAAACA UCAAGGGUUG GUGUCGGCAG CCACUAGUCA UAA | $3.93E-01$ |
| YLHHRz_1mut | AUAUACCCGU CUUCCCUUGC AAUCCAAAAA AAUCUGAUGA AGGAGAAACG AGGCAUGGCU UCGUCGGAAG ACGUUAUCAA AGA | $7.72E-01$ |
| YLHHRz_2mute | AUAGUGUCGU CUUCCCCCAU AAUCCAAAGA GAUUUGAUGA AGGAGACACG GAGCAUGGCU UCGUCGGAAG ACGUUAUCAA AGA | $8.02E-01$ |
| Design template | oooooooooo ooooCCUGAU GAoooooooo oooooooooo AGGAGAAAoo ooooAUGGCU UCGUCGGAAG ACGUUAUCAA AGA | − |
| Secondary structure | .......((( (((((..... .(((...... ..[[[[[[[ ..)))..((( ((((...))) )))).))))) )))]]]]]]] ].. | − |
| | glmS ribozyme RNA sequences | |

| | | |
|---|---|---|
| glmS_1 | AGCGCCAGCU CUAUGUCAGA AAAAAAAAGC UGACAUGGAG GACGAGGGCC CAGCAAAUCG AAAAAUCGGC GGAAGCUGGG GGGGCAGUGC GGGCCCCAAA A**G**GCGACC**U**C AAA**U**AGGCGU GAAAACGCCU **A**AUACACGGU UGC**C**ACCGCA CA | $6.96E-02$ |
| glmS_2 | AGCGCCAGGU CGCGUCUAUA AGGUUAAAAG UAGGCGCGAC GACGAGGGCC AGCCAAAUCG AGACAUCGGC GGGAGGCUGG GGGCCGGGUG UGGGUCCAGA GAGGUGGUGC AAAACAGGGG GAAACUCCUG CAUAAAAGCC ACCGACGCAC UC | $6.36E-02$ |
| Wild type | AGCGCCUGGA CUUAAAGCCA UUGCACUCCG GCUUUAAGUU GACGAGGGCA GGGUUUAUCG AGACAUCGGC GGGUGCCCUG CGGUCUUCCU GCGACCGUUA GAGGACUGUG AAAACCACAG GCGACUGUGG CAUAGAGCAG UCCGGGCAGG AA | $3.37E-01$ |
| Mutant | AAAACCUGGA AUUAAAGCCA UUGCACUCCG GCUUUAAGUU GACGAGGGCA GGGUUUAUCG AGACAUCGGC GGGUGCCCUG CGGUCUUCCU GCGACCGUUA GAGGACUGUG AAAACCACAG GCGACUGUGG CAUAGAGCAG UCCGGGCAGG AA | $3.11E-01$ |
| Design template | AGCGCCoGoo Cooooooooo oooooooooo oooooooGoo GACGAGGooo ooooooAUCG AGACAUCGGC GGRoGooooo oGGooooooo oooooCoooo oAooooooGo AAAAoooooo GoRAoooooo CAoAoAoooo oooGoooooo oo | – |
| Secondary structure | ..[[[[.[(( (((((((((. .........) )))))))))) ..[[[...(( ((((...]]] (....)]]]] ]...)))))) (((((.[[[[ [[)))))... ..((((((.. .(..(((((( ....)))))) ..)....))) )))..]]]]] ]. | – |

Table 3: RNA sequence data generated by Enzymer for all ribozymes

| **Annotation** | **Designed DNA sequences** |
|---|---|
| YLHHRz_1_GFP | TTTACACTTT ATGCTTCCGG CTCGTATGTT ATAGTGTCGT CTTC**CCTGAT** **GA**TCCAAAGA GATTTGAUGA AGGAGAAACG GAGCAUGGCT TCGTCGGAAG ACGTTATCAA AGA |
| YLHHRz_2_GFP | TTTACACTTT ATGCTTCCGG CTCGTATGTT ATATACCCGT CTTC**CCTGAT** **GA**TCCAAAAA AATTTGATGA AGGAGAAACG AGGCATGGCT TCGTCGGAAG ACGTTATCAA AGA |

|  |  |
|---|---|
| YLHHRz_1_GFP_mut | TTTACACTTT ATGCTTCCGG CTCGTATGTT ATAGTGTCGT CTTC**CCCCAT** **AA**TCCAAAGA GATTTGATGA AGGAGACACG GAGCATGGCT TCGTCGGAAG ACGTTATCAA AGA |
| YLHHRz_1_GFP_mut | TTTACACTTT ATGCTTCCGG CTCGTATGTT ATATACCCGT CTTC**CCTTGC** **AA**TCCAAAAA AATCTGATGA AGGAGAAACG AGGCATGGCT TCGTCGGAAG ACGTTATCAA AGA |

Table 4: Designed ribozyme DNA sequences including the Lac operon

## 6.7 Acknowledgment

Computations were made on the super computer Guillimin at McGill University, managed by Calcul Quebec and Compute Canada. The operation of this super computer is funded by the Canada Foundation for Innovation (CFI), Ministre de l'economie, de l'Innovation et des Exportations du Quebec (MEIE), RMGA and the Fonds de recherche du Quebec —Nature et technologies (FRQ-NT). Wet lab experiments became possible with generous assistance provided by Jonathan Perreault from centre INRS-institute Armand Frappier.

## 6.8 Author Contributions

The study was designed and formulated by Kasra Zandi (KZ), Dr. N.Kharma (NK) and Dr. J.Perreault (JP). The ribozymes were selected and designed by KZ. The computations were done and design templates and sequences were generated by KZ. The wet-lab experiments were carried and the experimental results including the related charts were generated by Sabrine Najeh (SN). The analysis was done by KZ, NJ and JP. This manuscript was written by KZ and SN. JP supervised the design, experimentation and interpretation process. JP and NK revised the final manuscript for submission. NK provided overall supervision and made the collaborative efforts possible. KZ and SN share first co-authorship and declare equal contribution.

# Chapter 7

# Conclusions and Future Work

We present a summary of our objectives and contributions. We describe the significance of this work in the field of computer science as well as bioinformatics and engineering of artificial nano structures with complex structural attributes. Finally, we propose directions for future studies.

## 7.1 Summary of thesis objectives

RNA structures such as ribozymes are attractive molecules as they are abundant in nature and have applications in several domains such as nano-engineering, therapeutics and synthetic biology. The first objective of this thesis was to devise a novel computational method for the design of functional RNA secondary structures. Computational methods for the design of RNA structures have numerous benefits when compared with experimental approaches. Computational design methods provide better flexibility and can perform the design task in shorter times and at significantly lower costs than the experimental methods. In this work, our main focus was to extend the class of structural features one can include in the design process by realizing an important structural feature called a *pseudoknot*. Pseudoknots are abundant in nature and are known to play key roles in stabilizing the functional forms of several different classes of ncRNAs such as ribozymes and riboswitches. The few existing methods that can handle pseudoknots result in low quality *in-silico* designs. Our aim was to improve on our current ability to design pseudoknots.

Despite availability of numerous computational methods for the design of RNA structures, only a few provide experimental evidence to support the applicability of their design approaches. Even fewer of them are capable of handling the more difficult cases where complex and important structural features such as pseudoknots are present. Without biological evidence, the applicability of computational methods remains questionable. A useful computational model is one that can produce high quality *in-silico* data and also shows validation of the results *in-vitro* or *in-vivo*. Beyond the development of a new and efficient computational method for the design of high quality pseudoknotted RNA structures, our second objective was to validate the applicability of our computational method *in-vitro*. Biological validation of the quality of computationally designed RNA molecules is an important milestone in the development of reliable molecular design and synthesis pipelines. A reliable RNA designer method, which can handle pseudoknots opens the door for wet-lab practitioners and bioinformaticians to design novel RNA structures with novel and valuable structural and functional attributes.

## 7.2 Summary of thesis contributions in computer science

RNA design can be modelled as a combinatorial optimization problem and is shown to be NP-hard. We needed to develop an efficient optimization process to solve the RNA design problem including pseudoknots. An important consideration is the quality. It turns out that existing pseudoknot designer methods use inferior quality measures. If one is required to optimize for significantly higher quality, then the fitness evaluation becomes an expensive task. For instance, it has been shown that the *ensemble defect* minimization gives the best results (Zadeh et al., 2011b). The problem is that computing the ensemble defect of a given RNA molecule which includes pseudoknots is in $\mathcal{O}(n^5)$ time (Dirks and Pierce, 2003). We needed to develop an optimization algorithm which can design pseudoknotted molecules with higher quality than our current reach while making sure designing diverse and large molecules remain tractable. It is desired to manage to design molecules that are as large as our existing energy models can characterize. Existing energy models can accurately model RNAs of up to 150 nucleotides in size.

Our first computational contribution is a new *adaptive defect weighted sampling* algorithm named `Enzymer` to solve the RNA design problem where pseudoknots can be included. The term adaptive means that during the optimization process the algorithm will dynamically decide how far in the mutational landscape the next solution candidate should be positioned relative to the current solution. This distance is calculated at each iteration as a function of the quality of the current candidate as approximated by the Boltzmann distribution. Using this technique the algorithm can jump from one corner of the mutational landscape to another far corner with the aim of improving the odds of finding better solutions. Our results showed our *adaptive* search technique leads to faster convergence hence reducing the number of times the expensive fitness function needs to be computed (section 5.5.3). The term *defect weighted sampling* means the probability of mutation of a position inside of a current solution, is proportional to the positional contribution of that position in the global ensemble defect of the molecule. We added an extra twist to the defect weighted sampling process. Our defect weighted sampling method, which is responsible for introducing positional mutations inside the solution candidate does not treat all positions inside of a candidate solution equally. The sampling method treats each position inside of the molecule according to the defectiveness of that position as specified by the Boltzmann distribution, which describes the quality of that position given its type (nested base pair v.s non-nested base pair v.s free base). Our data shows treating different positions of the molecular structure based on the type of the position introduced noticeable gain in convergence of the optimization process. Our benchmark dataset shows that the combination of our novel *adaptive* search and novel *defect weighted sampling* strategy leads to reduction in the number of times the fitness function is required to be computed (section 5.5.2). Our *adaptive defect weighted sampling* method makes it possible to reach quality levels much higher than those of other pseudoknotted RNA designer methods that implement the NSGA II and ACO algorithms (section 5.5.1).

Our second computational contribution is a design pipeline named `Enzymer-pipeline` to reengineer naturally occurring RNAs. This pipeline uses the catalytic core of functional RNAs, which may be extracted from multiple sequence alignment studies to initialize a design seed for our optimization algorithm. This pipeline allows one to describe a set of constraints as an input design template for the optimization process and let the algorithm derive the rest of the content by minimizing the ensemble defect. Enzymer is the name of our RNA design pipeline and makes use of our new optimization algorithm. Enzymer is implemented as a python 2.7 application and is publicly available. Our work including the proposed algorithm, the design pipeline and the software product has been published in the RNA section of the Frontiers in Genetics journal (impact factor 3.78) (Zandi et al., 2016).

## 7.3 Summary of thesis contributions in bioinformatics

RNA designer methods rely on energy models. Given a nucleotide composition, energy models describe the characteristics of RNA molecules at thermodynamic equilibrium. Energy models enable one to compute equilibrium attributes such as free energy, partition function, base pair probability and ensemble defect. Generally speaking, energy models are derived for specific contexts. For instance some models are tailored to model functional RNAs while some others are to characterize RNA shapes such as tiles and origamis. It is important to validate energy models in different contexts to obtain an accurate understanding about their applicability in various design contexts. On the other hand, the molecules designed by optimization algorithms which themselves use some energy model, need to be tested in experimental setups, so their functional and structural properties can be validated. Experimental validation is a testimony for usefulness of the designer computational method used as well as the applicability of the underlying model.

We presented our third contribution by using the Enzymer pipeline (Algorithm 9) to reengineer and test 18 novel artificial ribozyme sequences from three different species. The ribozyme species are one hammerhead ribozyme from mouse gut metagenome, one hammerhead ribozyme from the fungus *Yarrowia lipolytica* and one glmS ribozyme from *Thermoanaerobacter tengcongensis*. We tested all of the ribozymes *in-vitro* and showed that all except two were active (sections 6.4.1 to 6.4.3). To the best of our knowledge this is the first time a computational method has been used to design pseudoknotted functional RNAs followed by successful experimental validation. Our results demonstrate that Enzymer is a reliable tool for designing active ribozymes even when complex structural features such as pseudoknots are included. Moreover, our results show for the first time the underlying energy model of Dirks and Pierce (Dirks and Pierce, 2003) can be effectively applied to design active ribozymes. Our validated results bridges the gap between RNA design and experimental validation of complex and relatively large functional RNAs of up to 150 nucleotides. Our experimental results have been submitted to the RNA Journal (impact factor 4.94) and at the time of writing this document are under review (Zandi et al., 2018).

As our fourth contribution, we proposed a novel architecture for a ribozyme based gene regulatory network and tested it *in-vivo*. In our proposed architecture, the $3'$ end coding sequence of a self-cleaving hammerhead ribozyme from *Yarrowia lipolytica* was designed in such a way to overlap with the $5'$ end coding sequence of a reporter gene which in our case was the RFP protein. The transcription activation of the ribozyme was put under the control of the LacI DNA promoter which can be triggered by IPTG. In 7 out of 12 cases we observed the expected results. We observed that when the IPTG inducer was added, the expression of the GFP was intensified as expected. Our results allow one to hypothesize that by fine tuning this architecture, one may be able to design novel gene regulatory networks by combining ribozymes with protein coding sequences as a way to modulate the expression of the reporter gene. Our results were inconclusive hence, omitted from our journal submission.

## 7.4 Significance of our work

The final output of our work is an easy to use and publicly available software named Enzymer. Enzymer implements our new optimization algorithm and combines it with a design pattern for designing novel or reengineering naturally occurring functional RNA. Enzymer enables one to design novel RNA sequences with targeted secondary structures that include pseudoknots with quality higher than those of the best leading edge approaches. Notably our designed ribozymes were experimentally tested and almost all of the cases proved to be effective. The diverse sequence composition and quality that can be achieved by Enzymer gives wet-lab experimentalists the means to study RNA molecules and their associated sequence-structure attributes such as ensemble defect, free energy, mutational robustness, and molecular plasticity. Our work opens the door to the reliable and efficient design of complex ribozymes and other RNAs with given, possibly novel functionalities.

## 7.5 Areas for future studies

Interesting areas for future exploration from a computer science prospective may include development of multi-objective optimization methods for the design of functional RNAs that can fold in more than a single secondary structure. Multi-target design is particularly useful when designing RNA switches such as riboswitches. On another point, recent developments (Wolfe and Pierce, 2014) have made it possible to compute the partition function of a test tube of interacting

RNA strands. An interesting area of future research is to develop sequence optimizers for targeted RNA-RNA interactions allowing design of complex RNA based circuits.

Areas for further research and lab work may include large scale studies to characterize the relationships between the design temperature and testing temperature. In our experimental validation we designed the ribozyme sequences for optimal folding at various temperatures, however only tested them all at the same temperature. Although our current energy models are not able to characterize equilibrium characteristics at varying salt conditions, varying the design temperature may be an effective way to design for varying salt conditions. It will be useful to observe the sensitivity of enzymatic activity of the designed ribozymes at varying design temperatures, varying test temperatures as well as varying salt conditions. Large scale *in-vitro* testing will allow one to perform such sensitivity analyses.

It remains in our interest to further understand the sensitivity of functionality of the RNA structures relative to different quality measures such as folding energy and ensemble defect. The normalized ensemble defect of the ribozymes we tested were varying between 5% to 10% however there was no clear signal in our data to correlate the ensemble defect value to cleavage. It will be an interesting study to deign and test ribozyme sequences with varying ensemble defect to understand the relationship between cleavage and various sequence quality values. We suspect it may be possible for ribozyme sequences that have a higher ensemble defect to still remain highly functional. If this hypothesis turns out to be correct, one can save significant amounts of computational time by adjusting the stop criteria during the sequence optimization process. The relationship between the ensemble defect, sequence composition and catalytic activity requires further investigation.

Finally, despite our *in-vivo* results for the gene regulatory construct which we proposed were inconclusive, we believe it is worth fine tunning the design of the proposed construct. One may move around the position of the ribosome binding site or tweak the size of the overlap between the reporter gene sequence and the ribozyme sequence. Another potent tweak would be using different hammerhead ribozyme species than the one used in the original design. It is desirable to characterize the minimal set of sequence and structural requirements which would lead to a robust gene regulatory architecture similar to the one we proposed in Chapter 6.

# Bibliography

Afonin, K. A., Bindewald, E., Yaghoubian, A. J., Voss, N., Jacovetty, E., Shapiro, B. A., and Jaeger, L. (2010). *In-vitro* assembly of cubic RNA-based scaffolds designed in silico. *Nature nanotechnology*, 5(9):676–682.

Afonin, K. A., Cieply, D. J., and Leontis, N. B. (2008a). Specific RNA self-assembly with minimal paranemic motifs. *Journal of the American Chemical Society*, 130(1):93–102.

Afonin, K. A., Danilov, E. O., Novikova, I. V., and Leontis, N. B. (2008b). TokenRNA: A new type of sequence-specific, label-free fluorescent biosensor for folded RNA molecules. *Chembiochem*, 9(12):1902–1905.

Afonin, K. A., Lindsay, B., and Shapiro, B. A. (2013). Engineered RNA nanodesigns for applications in RNA nanotechnology. *RNA Nanotechnology*, 1:1–15.

Aguirre, J., Buldú, J. M., Stich, M., and Manrubia, S. C. (2011). Topological structure of the space of phenotypes: the case of RNA neutral networks. *PloS One*, 6(10):e26324.

Aherne, D., Gara, M., Kelly, J. M., and Gun'ko, Y. K. (2010). From Ag nanoprisms to triangular AuAg nanoboxes. *Advanced Functional Materials*, 20(8):1329–1338.

Aissi, H., Bazgan, C., and Vanderpooten, D. (2009). Min–max and min–max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197(2):427–438.

Akutsu, T. (2000). Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*, 104(1):45–62.

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838.

Amato, F., López, A., Peña-Méndez, E. M., Vaňhara, P., Hampl, A., and Havel, J. (2013). Artificial neural networks in medical diagnosis.

Ambros, V. (2004). The functions of animal microRNAs. *Nature*, 431(7006):350–355.

Andersen, F. F., Knudsen, B., Oliveira, C. L. P., Frøhlich, R. F., Krüger, D., Bungert, J., Agbandje-McKenna, M., McKenna, R., Juul, S., Veigaard, C., et al. (2008). Assembly and structural analysis of a covalently closed nano-scale DNA cage. *Nucleic Acids Research*, 36(4):1113–1119.

Andronescu, M., Bereg, V., Hoos, H. H., and Condon, A. (2008). RNA STRAND: the RNA secondary structure and statistical analysis database. *BMC bioinformatics*, 9(1):340.

Andronescu, M., Fejes, A. P., Hutter, F., Hoos, H. H., and Condon, A. (2004). A new algorithm for RNA secondary structure design. *Journal of Molecular Biology*, 336(3):607–624.

Archetti, C., Bertazzi, L., Laporte, G., and Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391.

Ausländer, S., Ketzer, P., and Hartig, J. S. (2010). A ligand-dependent hammerhead ribozyme switch for controlling mammalian gene expression. *Molecular BioSystems*, 6(5):807–814.

Avihoo, A., Churkin, A., and Barash, D. (2011). RNAexinv: An extended inverse RNA folding from shape and physical attributes to sequences. *BMC bioinformatics*, 12(1):319.

Babaioff, M., Immorlica, N., Kempe, D., and Kleinberg, R. (2007). A knapsack secretary problem with applications. *Approximation, randomization, and combinatorial optimization. Algorithms and techniques*, pages 16–28.

Baker, J. L., Sudarsan, N., Weinberg, Z., Roth, A., Stockbridge, R. B., and Breaker, R. R. (2012). Widespread genetic switches and toxicity resistance proteins for fluoride. *Science*, 335(6065):233–235.

Baker, T., Gill, J., and Solovay, R. (1975). Relativizations of the P=NP question. *SIAM Journal on computing*, 4(4):431–442.

Ban, N., Freeborn, B., Nissen, P., Penczek, P., Grassucci, R. A., Sweet, R., Frank, J., Moore, P. B., and Steitz, T. A. (1998). A 9 Å resolution x-ray crystallographic map of the large ribosomal subunit. *Cell*, 93(7):1105–1115.

Bartel, D. P. (2009). MicroRNAs: target recognition and regulatory functions. *Cell*, 136(2):215–233.

Barth, G. and Gaillardin, C. (1997). Physiology and genetics of the dimorphic fungus *Yarrowia lipolytica*. *FEMS Microbiology Reviews*, 19(4):219–237.

Bell, J. E. and McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced engineering informatics*, 18(1):41–48.

Bellaousov, S. and Mathews, D. H. (2010). ProbKnot: fast prediction of RNA secondary structure including pseudoknots. *RNA*, 16(10):1870–1880.

Beni, G. and Wang, J. (1993). Swarm intelligence in cellular robotic systems. In *Robots and Biological Systems: Towards a New Bionics?*, pages 703–712. Springer.

Bhartiya, D. and Scaria, V. (2016). Genomic variations in non-coding RNAs: structure, function and regulation. *Genomics*, 107(2):59–68.

Bianchi, L., Dorigo, M., Gambardella, L. M., and Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing: an international journal*, 8(2):239–287.

Bindewald, E., Afonin, K., Jaeger, L., and Shapiro, B. A. (2011). Multistrand RNA secondary structure prediction and nanostructure design including pseudoknots. *ACS nano*, 5(12):9542–9551.

Bindewald, E., Grunewald, C., Boyle, B., OâĂŹConnor, M., and Shapiro, B. A. (2008). Computational strategies for the automated design of RNA nanoscale structures from building blocks using NanoTiler. *Journal of Molecular Graphics and Modelling*, 27(3):299–308.

Bland, R. G. and Shallcross, D. F. (1989). Large travelling salesman problems arising from experiments in X-ray crystallography: a preliminary report on computation. *Operations Research Letters*, 8(3):125–128.

Blazewicz, J., Szachniuk, M., and Wojtowicz, A. (2005). RNA tertiary structure determination: NOE pathways construction by tabu search. *Bioinformatics*, 21(10):2356–2361.

Blum, C. and Dorigo, M. (2004). The hyper-cube framework for ant colony optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(2):1161–1172.

Blum, C. and Sampels, M. (2004). An ant colony optimization algorithm for shop scheduling problems. *Journal of Mathematical Modelling and Algorithms*, 3(3):285–308.

Bodini, O. and Ponty, Y. (2010). Multi-dimensional boltzmann sampling of languages. *arXiv preprint arXiv:1002.0046*.

Bonnet, E., Rzazewski, P., and Sikora, F. (2017). Designing RNA secondary structures is NP-Hard. *arXiv preprint arXiv:1710.11513*.

Bratkovič, T. and Rogelj, B. (2014). The many faces of small nucleolar RNAs. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, 1839(6):438–443.

Breaker, R. R. (2012). Riboswitches and the RNA world. *Cold Spring Harbor perspectives in biology*, 4(2):a003566.

Brierley, I., Pennell, S., and Gilbert, R. J. (2007). Viral RNA pseudoknots: versatile motifs in gene expression and replication. *Nature Reviews Microbiology*, 5(8):598–610.

Brophy, J. A. and Voigt, C. A. (2014). Principles of genetic circuit design. *Nature methods*, 11(5):508–520.

Bullnheimer, B., Hartl, R. F., and Strauss, C. (1997). A new rank based version of the ant system. a computational study. *SFB Adaptive Information Systems and Modelling in Economics and Management Science*.

Burkard, R. E. and Rendl, F. (1984). A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operational Research*, 17(2):169–174.

Burke, E. and Silva, J. L. (2005). The design of memetic algorithms for scheduling and timetabling problems. In *Recent Advances in Memetic Algorithms*, pages 289–311. Springer.

Burnett, J. C. and Rossi, J. J. (2012). RNA-based therapeutics: current progress and future prospects. *Chemistry and Biology*, 19(1):60–71.

Busch, A. and Backofen, R. (2006). INFO-RNA: a fast approach to inverse RNA folding. *Bioinformatics*, 22(15):1823–1831.

Byun, Y. and Han, K. (2009). PseudoViewer3: generating planar drawings of large-scale RNA structures with pseudoknots. *Bioinformatics*, 25(11):1435–1437.

Cameron, D. E., Bashor, C. J., and Collins, J. J. (2014). A brief history of synthetic biology. *Nature Reviews Microbiology*, 12(5):381–390.

Campbell, A. M. and Savelsbergh, M. (2004). Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science*, 38(3):369–378.

Cao, S. and Chen, S.-J. (2011). Structure and stability of RNA/RNA kissing complex: with application to HIV dimerization initiation signal. *RNA*, 17(12):2130–2143.

Çavuşlar, G., Çatay, B., and Apaydın, M. S. (2012). A tabu search approach for the NMR protein structure-based assignment problem. *IEEE/ACM transactions on computational biology and bioinformatics*, 9(6):1621–1628.

Chang, T.-H., Huang, H.-Y., Hsu, J. B.-K., Weng, S.-L., Horng, J.-T., and Huang, H.-D. (2013). An enhanced computational platform for investigating the roles of regulatory RNA and for identifying functional RNA motifs. *BMC bioinformatics*, 14(2):1.

Chappell, J., Watters, K. E., Takahashi, M. K., and Lucks, J. B. (2015). A renaissance in rna synthetic biology: new mechanisms, applications and tools for the future. *Current opinion in chemical biology*, 28:47–56.

Cheeseman, P. C., Kanefsky, B., and Taylor, W. M. (1991). Where the really hard problems are. In *IJCAI*, volume 1, pages 331–337.

Chen, H.-L., Condon, A., and Jabbari, H. (2009). An $\mathcal{O}(n^5)$ algorithm for MFE prediction of kissing hairpins and 4-chains in nucleic acids. *Journal of Computational Biology*, 16(6):803–815.

Chen, J.-H., Le, S.-Y., and Maizel, J. V. (2000). Prediction of common secondary structures of RNAs: a genetic algorithm approach. *Nucleic Acids Research*, 28(4):991–999.

Chen, Y. Y., Jensen, M. C., and Smolke, C. D. (2010). Genetic control of mammalian T-cell proliferation with synthetic RNA regulatory systems. *Proceedings of the National Academy of Sciences*, 107(19):8531–8536.

Cheng, R., Gen, M., and Tsujimura, Y. (1996). A tutorial survey of job-shop scheduling problems using genetic algorithms. *Computers & industrial engineering*, 30(4):983–997.

Cheung, K. L. and Fu, A. W.-C. (1998). Enhanced nearest neighbour search on the R-tree. *ACM SIGMOD Record*, 27(3):16–21.

Chipot, C. and Pohorille, A. (2007). Free energy calculations. *Verlag Berlin Heidelberg*.

Churkin, A., Retwitzer, M. D., Reinharz, V., Ponty, Y., Waldispühl, J., and Barash, D. (2017). Design of RNAs: comparing programs for inverse RNA folding. *Briefings in bioinformatics*, page bbw120.

Chvatal, V. (1979). A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235.

Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A., et al. (2007). *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer.

Collins, J. A., Irnov, I., Baker, S., and Winkler, W. C. (2007). Mechanism of mRNA destabilization by the glmS ribozyme. *Genes & development*, 21(24):3356–3368.

Condon, A., Davy, B., Rastegari, B., Zhao, S., and Tarrant, F. (2004). Classifying RNA pseudoknotted structures. *Theoretical Computer Science*, 320(1):35–50.

Custódio, F. L., Barbosa, H. J., and Dardenne, L. E. (2014). A multiple minima genetic algorithm for protein structure prediction. *Applied Soft Computing*, 15:88–99.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314.

Dantzig, G. (2016). *Linear programming and extensions*. Princeton University Press.

Darwin, C. and Bynum, W. F. (2009). *The origin of species by means of natural selection: or, the preservation of favored races in the struggle for life*. Penguin.

Das, R., Karanicolas, J., and Baker, D. (2010). Atomic accuracy in predicting and designing noncanonical RNA structure. *Nature methods*, 7(4):291–294.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International Conference on Parallel Problem Solving From Nature*, pages 849–858. Springer.

Delebecque, C. J., Lindner, A. B., Silver, P. A., and Aldaye, F. A. (2011). Organization of intracellular reactions with rationally designed RNA assemblies. *Science*, 333(6041):470–474.

Dibrov, S. M., McLean, J., Parsons, J., and Hermann, T. (2011). Self-assembling RNA square. *Proceedings of the National Academy of Sciences*, 108(16):6405–6408.

Dieterich, C. and Stadler, P. F. (2013). Computational biology of RNA interactions. *Wiley Interdisciplinary Reviews: RNA*, 4(1):107–120.

Ding, S.-W. (2010). RNA-based antiviral immunity. *Nature reviews. Immunology*, 10(9):632.

Ding, Y., Chan, C. Y., and Lawrence, C. E. (2004). Sfold web server for statistical folding and rational design of nucleic acids. *Nucleic Acids Research*, 32(suppl 2):W135–W141.

Ding, Y., Chan, C. Y., and Lawrence, C. E. (2005). RNA secondary structure prediction by centroids in a boltzmann weighted ensemble. *RNA*, 11(8):1157–1166.

Ding, Y. and Lawrence, C. E. (1999). A bayesian statistical algorithm for RNA secondary structure prediction. *Computers and Chemistry*, 23(3):387–400.

Ding, Y. and Lawrence, C. E. (2003). A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Research*, 31(24):7280–7301.

Dirks, R. M., Lin, M., Winfree, E., and Pierce, N. A. (2004). Paradigms for computational nucleic acid design. *Nucleic Acids Research*, 32(4):1392–1403.

Dirks, R. M. and Pierce, N. A. (2003). A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of computational chemistry*, 24(13):1664–1677.

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., and Gingeras, T. R. (2013). Star: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21.

Dock-Bregeon, A., Chevrier, B., Podjarny, A., Johnson, J., De Bear, J., Gough, G., Gilham, P., and Moras, D. (1989). Crystallographic structure of an RNA helix. *Journal of molecular biology*, 209(3):459–474.

Dorigo, M., Birattari, M., and Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39.

Dorigo, M. and Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 2, pages 1470–1477. IEEE.

Dorigo, M. and Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1):53–66.

Dorigo, M., Maniezzo, V., and Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41.

Dotu, I., Garcia-Martin, J. A., Slinger, B. L., Mechery, V., Meyer, M. M., and Clote, P. (2014). Complete RNA inverse folding: computational design of functional hammerhead ribozymes. *Nucleic Acids Research*, pages 740–752.

Dotu, I., Lorenz, W. A., Van Hentenryck, P., and Clote, P. (2009). Computing folding pathways between RNA secondary structures. *Nucleic Acids Research*, 38(5):1711–1722.

Ducatelle, F., Di Caro, G. A., and Gambardella, L. M. (2010). Principles and applications of swarm intelligence for adaptive routing in telecommunications networks. *Swarm Intelligence*, 4(3):173–198.

Duchon, P., Flajolet, P., Louchard, G., and Schaeffer, G. (2004). Boltzmann samplers for the random generation of combinatorial structures. *Combinatorics, Probability and Computing*, 13(4-5):577–625.

Dueck, G. and Scheuer, T. (1990). Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. *Journal of computational physics*, 90(1):161–175.

Eddy, S. R. (2004). How do RNA folding algorithms work? *Nature biotechnology*, 22(11):1457–1458.

Edelstein, M. L., Abedi, M. R., and Wixon, J. (2007). Gene therapy clinical trials worldwide 2007: an update. *The journal of gene medicine*, 9(10):833–842.

Edwards, A. L., Reyes, F. E., Héroux, A., and Batey, R. T. (2010). Structural basis for recognition of S-adenosylhomocysteine by riboswitches. *RNA*, 16(11):2144–2155.

Eiben, A. E., Aarts, E. H., and Van Hee, K. M. (1990). Global convergence of genetic algorithms: A Markov chain analysis. In *International Conference on Parallel Problem Solving from Nature*, pages 3–12. Springer.

Ekren, O. and Ekren, B. Y. (2010). Size optimization of a PV/wind hybrid energy conversion system with battery storage using simulated annealing. *Applied Energy*, 87(2):592–598.

Elbaz, J. and Willner, I. (2012). DNA origami: nanorobots grab cellular control. *Nature materials*, 11(4):276–277.

Engelbrecht, A. P. (2006). *Fundamentals of computational swarm intelligence*. John Wiley & Sons.

Esmaili-Taheri, A. and Ganjtabesh, M. (2015). ERD: a fast and reliable tool for RNA design including constraints. *BMC bioinformatics*, 16(1):20.

Esposito, C. L., Catuogno, S., and De Franciscis, V. (2014). Aptamer-mediated selective delivery of short RNA therapeutics in cancer cells. *Journal of RNAi and gene silencing: an international journal of RNA and gene targeting research*, 10:500.

Fatica, A. and Bozzoni, I. (2014). Long non-coding RNAs: new players in cell differentiation and development. *Nature Reviews Genetics*, 15(1):7–21.

Feigon, J. (2015). Back to the future of RNA structure. *RNA*, 21(4):611–612.

Fellmuth, B., Gaiser, C., and Fischer, J. (2006). Determination of the Boltzmann constantâĂŤstatus and prospects. *Measurement Science and Technology*, 17(10):R145.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181.

Findeiss, S., Wachsmuth, M., Morl, M., and Stadler, P. F. (2015). Chapter One-Design of Transcription Regulating Riboswitches. *Methods in Enzymology*, 550:1–22.

Fisher, M. L. (1981). The Lagrangian relaxation method for solving integer programming problems. *Management science*, 27(1):1–18.

Flajolet, P., Fusy, É., and Pivoteau, C. (2007). Boltzmann sampling of unlabelled structures. In *Proceedings of the Meeting on Analytic Algorithmics and Combinatorics*, pages 201–211. Society for Industrial and Applied Mathematics.

Flood, M. M. (1956). The traveling-salesman problem. *Operations Research*, 4(1):61–75.

Fogel, D. B. (1992). Evolving artificial intelligence.

Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). Artificial intelligence through simulated evolution. 1.

Fortnow, L. (2009). The status of the P versus NP problem. *Communications of the ACM*, 52(9):78–86.

Freier, S. M., Kierzek, R., Jaeger, J. A., Sugimoto, N., Caruthers, M. H., Neilson, T., and Turner, D. H. (1986). Improved free-energy parameters for predictions of RNA duplex stability. *Proceedings of the National Academy of Sciences*, 83(24):9373–9377.

Freisleben, B. and Merz, P. (1996). New genetic local search operators for the traveling salesman problem. *Parallel Problem Solving from Nature (PPSN) IV*, pages 890–899.

Freitag, D. (2017). Greedy attribute selection. In *Machine Learning Proceedings 1994: Proceedings of the Eighth International Conference*, page 28. Morgan Kaufmann.

Frommer, J., Appel, B., and Muller, S. (2015). Ribozymes that can be regulated by external stimuli. *Current opinion in biotechnology*, 31:35–41.

Fu, J. and Yan, H. (2012). Controlled drug release by a nanorobot. *Nature biotechnology*, 30(5):407–408.

Fu, Y., Xu, Z., Lu, Z. J., Zhao, S., and Mathews, D. H. (2013). 31 Discovery of novel ncRNA by scanning multiple genome alignments. *Journal of Biomolecular Structure and Dynamics*, 31(sup1):19–19.

Gao, J. Z., Li, L. Y., and Reidys, C. M. (2010). Inverse folding of RNA pseudoknot structures. *Algorithms for Molecular Biology*, 5(1):1.

Garcia-Martin, J. A., Clote, P., and Dotu, I. (2013). RNAiFOLD: a constraint programming algorithm for RNA inverse folding and molecular design. *Journal of bioinformatics and computational biology*, 11(02):1350001.

Gardiner, E. J., Willett, P., and Artymiuk, P. J. (2001). Protein docking using a genetic algorithm. *Proteins: Structure, Function, and Bioinformatics*, 44(1):44–56.

Gardner, P. P., Daub, J., Tate, J. G., Nawrocki, E. P., Kolbe, D. L., Lindgreen, S., Wilkinson, A. C., Finn, R. D., Griffiths-Jones, S., Eddy, S. R., et al. (2009). Rfam: updates to the RNA families database. *Nucleic Acids Research*, 37(suppl 1):D136–D140.

Garey, M. R., Johnson, D. S., and Stockmeyer, L. (1976). Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267.

Garibotti, A. V., Liao, S., and Seeman, N. C. (2007). A simple DNA-based translation system. *Nano letters*, 7(2):480–483.

Geary, C., Rothemund, P. W., and Andersen, E. S. (2014). A single-stranded architecture for cotranscriptional folding of RNA nanostructures. *Science*, 345(6198):799–804.

Geem, Z. W., Kim, J. H., and Loganathan, G. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2):60–68.

Gendreau, M., Iori, M., Laporte, G., and Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science*, 40(3):342–350.

Germer, K., Leonard, M., and Zhang, X. (2013). RNA aptamers and their therapeutic and diagnostic applications. *Int J Biochem Mol Biol*, 4(1):27–40.

Gibbs, M. J., Armstrong, J. S., and Gibbs, A. J. (2000). Sister-scanning: a Monte Carlo procedure for assessing signals in recombinant sequences. *Bioinformatics*, 16(7):573–582.

Giegé, R., Puglisi, J. D., and Florentz, C. (1993). tRNA structure and aminoacylation efficiency. *Progress in nucleic acid research and molecular biology*, 45:129–206.

Gilbert, S. D., Rambo, R. P., Van Tyne, D., and Batey, R. T. (2008). Structure of the SAM-II riboswitch bound to S-adenosylmethionine. *Nature structural & molecular biology*, 15(2):177–182.

Glover, F. and Laguna, M. (2013). Tabu Search. In *Handbook of Combinatorial Optimization*, pages 3261–3362. Springer.

Goodman, R. P., Heilemann, M., Doose, S., Erben, C. M., Kapanidis, A. N., and Turberfield, A. J. (2008). Reconfigurable, braced, three-dimensional DNA nanostructures. *Nature nanotechnology*, 3(2):93–96.

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

Grabow, W. W., Afonin, K. A., Zakrevsky, P., Walker, F. M., Calkins, E. R., Geary, C., Kasprzak, W., Bindewald, E., Shapiro, B. A., and Jaeger, L. (2012a). RNA nanotechnology in nanomedicine. *Nanomedicine and Drug Delivery*, 1:208–221.

Grabow, W. W. and Jaeger, L. (2014). RNA self-assembly and RNA nanotechnology. *Accounts of chemical research*, 47(6):1871–1880.

Grabow, W. W., Zhuang, Z., Swank, Z. N., Shea, J.-E., and Jaeger, L. (2012b). The right angle (RA) motif: a prevalent ribosomal RNA structural pattern found in group I introns. *Journal of Molecular Biology*, 424(1):54–67.

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, 5:287–326.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on*, pages 6645–6649. IEEE.

Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S. R., and Bateman, A. (2005). Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Research*, 33(suppl 1):D121–D124.

Grosan, C. and Abraham, A. (2007). Hybrid evolutionary algorithms: methodologies, architectures, and reviews. In *Hybrid evolutionary algorithms*, pages 1–17. Springer.

Gultyaev, A. P., Van Batenburg, F., and Pleij, C. W. (1995). The computer simulation of RNA folding pathways using a genetic algorithm. *Journal of molecular biology*, 250(1):37–51.

Guo, P. (2010). The emerging field of RNA nanotechnology. *Nature nanotechnology*, 5(12):833–842.

Gurtan, A. M. and Sharp, P. A. (2013). The role of miRNAs in regulating gene expression networks. *Journal of Molecular Biology*, 425(19):3582–3600.

Haleš, J., Maňuch, J., Ponty, Y., and Stacho, L. (2015). Combinatorial RNA design: Designability and Structure-Approximating Algorithm. In *Annual Symposium on Combinatorial Pattern Matching*, pages 231–246. Springer.

Hamada, M., Kiryu, H., Sato, K., Mituyama, T., and Asai, K. (2009). Prediction of RNA secondary structure using generalized centroid estimators. *Bioinformatics*, 25(4):465–473.

Hammann, C., Luptak, A., Perreault, J., and De La Peña, M. (2012). The ubiquitous hammerhead ribozyme. *RNA*, 18(5):871–885.

Hannon, G. J. (2002). RNA interference. *Nature*, 418(6894):244–251.

Hansen, P. and Mladenović, N. (2014). Variable neighborhood search. In *Search methodologies*, pages 313–337. Springer.

Harris, K. A., Lünse, C. E., Li, S., Brewer, K. I., and Breaker, R. R. (2015). Biochemical analysis of pistol self-cleaving ribozymes. *RNA*, 21(11):1852–1858.

Held, M. and Karp, R. M. (1970). The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18(6):1138–1162.

Hendrix, D. K., Brenner, S. E., and Holbrook, S. R. (2005). RNA structural motifs: building blocks of a modular biomolecule. *Quarterly reviews of biophysics*, 38(03):221–243.

Hochbaum, D. S. (1982). Approximation algorithms for the set covering and vertex cover problems. *SIAM Journal on computing*, 11(3):555–556.

Hochbaum, D. S. (1996). *Approximation algorithms for NP-hard problems.* PWS Publishing Co.

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

Hofacker, I. L., Fekete, M., and Stadler, P. F. (2002). Secondary structure prediction for aligned RNA sequences. *Journal of Molecular Biology*, 319(5):1059–1066.

Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, L. S., Tacker, M., and Schuster, P. (1994). Fast folding and comparison of RNA secondary structures. *Monatshefte für Chemie/Chemical Monthly*, 125(2):167–188.

Hoff, D. J. and Olver, P. J. (2014). Automatic solution of jigsaw puzzles. *Journal of mathematical imaging and vision*, 49(1):234–250.

Hohng, S., Wilson, T. J., Tan, E., Clegg, R. M., Lilley, D. M., and Ha, T. (2004). Conformational flexibility of four-way junctions in RNA. *Journal of molecular biology*, 336(1):69–79.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* MIT Press.

Hong, C. A. and Nam, Y. S. (2014). Functional nanostructures for effective delivery of small interfering RNA therapeutics. *Theranostics*, 4(12):1211–32.

Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2006). Automata theory, languages, and computation. *International Edition*, 24.

Huang, F. W., Peng, W. W., and Reidys, C. M. (2009). Folding 3-noncrossing RNA pseudoknot structures. *Journal of Computational Biology*, 16(11):1549–1575.

Huntzinger, E. and Izaurralde, E. (2011). Gene silencing by microRNAs: contributions of translational repression and mRNA decay. *Nature Reviews Genetics*, 12(2):99–111.

Isaacs, F. J., Dwyer, D. J., and Collins, J. J. (2006). RNA synthetic biology. *Nature biotechnology*, 24(5):545–554.

Ishwaran, H. and James, L. F. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173.

Jackson, R. N., McCoy, A. J., Terwilliger, T. C., Read, R. J., and Wiedenheft, B. (2015). X-ray structure determination using low-resolution electron microscopy maps for molecular replacement. *Nature protocols*, 10(9):1275–1284.

Janssen, S. and Giegerich, R. (2014). The RNA shapes studio. *Bioinformatics*, 31:btu649, doi:10.1093/bioinformatics/btu649.

Johnson, D. S., Papadimitriou, C. H., and Yannakakis, M. (1988). How easy is local search? *Journal of computer and system sciences*, 37(1):79–100.

Johnson, O. and Liu, J. (2006). A traveling salesman approach for predicting protein functions. *Source Code for Biology and Medicine*, 1(1):3.

Jones, M. R., Seeman, N. C., and Mirkin, C. A. (2015). Programmable materials and the nature of the DNA bond. *Science*, 347(6224):1260901.

Jonikas, M. A., Radmer, R. J., Laederach, A., Das, R., Pearlman, S., Herschlag, D., and Altman, R. B. (2009). Coarse-grained modelling of large RNA molecules with knowledge-based potentials and structural filters. *RNA*, 15(2):189–199.

Kanj, R., Joshi, R., and Nassif, S. (2006). Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events. In *Design Automation Conference, 2006 43rd ACM/IEEE*, pages 69–72. IEEE.

Kannan, A. A., Mao, G., and Vucetic, B. (2005). Simulated annealing based localization in wireless sensor network. In *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, pages 2–pp. IEEE.

Karaboga, D. and Akay, B. (2009). A survey: algorithms simulating bee swarm intelligence. *Artificial intelligence review*, 31(1-4):61–85.

Karaboga, D. and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, 39(3):459–471.

Kazakovtsev, L. A. and Antamoshkin, A. N. (2014). Genetic algorithm with fast greedy heuristic for clustering and location problems. *Informatica*, 38(3).

Kellerer, H. and Strusevich, V. A. (2010). Fully polynomial approximation schemes for a symmetric quadratic knapsack problem and its scheduling applications. *Algorithmica*, 57(4):769–795.

Kennedy, J. (2006). Swarm intelligence. In *Handbook of nature-inspired and innovative computing*, pages 187–219. Springer.

Khalil, A. S. and Collins, J. J. (2010). Synthetic biology: applications come of age. *Nature Reviews Genetics*, 11(5):367–379.

Kharma, N., Varin, L., Abu-Baker, A., Ouellet, J., Najeh, S., Ehdaeivand, M.-R., Belmonte, G., Ambri, A., Rouleau, G., and Perreault, J. (2016). Automated design of hammerhead ribozymes and validation by targeting the PABPN1 gene transcript. *Nucleic Acids Research*, 44(4):e39–e39.

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., et al. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

Klein, D., Schmeing, T., Moore, P., and Steitz, T. (2001). The kink-turn: a new RNA secondary structure motif. *The EMBO Journal*, 20(15):4214–4221.

Klein, D. J. and Ferré-D'Amaré, A. R. (2006). Structural basis of glmS ribozyme activation by glucosamine-6-phosphate. *Science*, 313(5794):1752–1756.

Kleinkauf, R., Houwaart, T., Backofen, R., and Mann, M. (2015). antaRNA:Multi-objective inverse folding of pseudoknot RNA using ant-colony optimization. *BMC bioinformatics*, 16(1):389.

Knowles, J. D. and Corne, D. W. (2000). M-PAES: A memetic algorithm for multiobjective optimization. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 325–332. IEEE.

Kole, R., Krainer, A. R., and Altman, S. (2012). RNA therapeutics: beyond RNA interference and antisense oligonucleotides. *Nature reviews drug discovery*, 11(2):125–140.

Korb, O., Stützle, T., and Exner, T. E. (2006). Plants: Application of ant colony optimization to structure-based drug design. In *International Workshop on Ant Colony Optimization and Swarm Intelligence*, pages 247–258. Springer.

Korupolu, M. R., Plaxton, C. G., and Rajaraman, R. (2000). Analysis of a local search heuristic for facility location problems. *Journal of algorithms*, 37(1):146–188.

Krasnogor, N. and Smith, J. (2000). A memetic algorithm with self-adaptive local search: TSP as a case study. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, pages 987–994. Morgan Kaufmann Publishers Inc.

Krishnanand, K. and Ghose, D. (2009). Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm intelligence*, 3(2):87–124.

Kritikos, M. and Ioannou, G. (2017). A greedy heuristic for the capacitated minimum spanning tree problem. *Journal of the Operational Research Society*, 68(10):1223–1235.

Krokhotin, A., Houlihan, K., and Dokholyan, N. V. (2015). iFoldRNA v2: folding RNA with constraints. *Bioinformatics*, page btv221.

Kunz, D. (1991). Channel assignment for cellular radio using neural networks. *IEEE Transactions on Vehicular Technology*, 40(1):188–193.

Kurniawan, T. B., Khalid, N. K., Ibrahim, Z., Khalid, M., and Middendorf, M. (2008). An ant colony system for DNA sequence design based on thermodynamics. In *Proceedings of the Fourth IASTED International Conference on Advances in Computer Science and Technology*, pages 144–149. ACTA Press.

Lainé, S., Scarborough, R. J., Lévesque, D., Didierlaurent, L., Soye, K. J., Mougel, M., Perreault, J.-P., and Gatignol, A. (2011). In vitro and in vivo cleavage of HIV-1 RNA by new SOFA-HDV ribozymes and their potential to inhibit viral replication. *RNA Biology*, 8(2):343–353.

Laing, C. and Schlick, T. (2011). Computational approaches to RNA structure prediction, analysis, and design. *Current opinion in structural biology*, 21(3):306–318.

Lancia, G., Carr, R., Walenz, B., and Istrail, S. (2001). 101 optimal PDB structure alignments: a branch-and-cut algorithm for the maximum contact map overlap problem. In *Proceedings of the fifth annual international conference on Computational biology*, pages 193–202. ACM.

Lau, M. W. and Ferré-D'Amaré, A. R. (2013). An in vitro evolved glmS ribozyme has the wild-type fold but loses coenzyme dependence. *Nature chemical biology*, 9(12):805–810.

Lau, M. W. and Ferré-D'Amaré, A. R. (2016). In vitro evolution of coenzyme-independent variants from the glmS ribozyme structural scaffold. *Methods*, 106:76–81.

Lau, M. W., Trachman, R. J., and Ferré-D'Amaré, A. R. (2017). A divalent cation-dependent variant of the glmS ribozyme with stringent $Ca_{2+}$ selectivity co-opts a preexisting nonspecific metal ion-binding site. *RNA*, 23(3):355–364.

Lawler, E. L. and Wood, D. E. (1966). Branch-and-bound methods: A survey. *Operations research*, 14(4):699–719.

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

Lenstra, J. K. and Kan, A. R. (1975). Some simple applications of the travelling salesman problem. *Journal of the Operational Research Society*, 26(4):717–733.

Leonard, J. N. and Schaffer, D. V. (2005). Computational design of antiviral RNA interference strategies that resist human immunodeficiency virus escape. *Journal of Virology*, 79(3):1645–1654.

Leontis, N. B., Lescoute, A., and Westhof, E. (2006). The building blocks and motifs of RNA architecture. *Current opinion in structural biology*, 16(3):279–287.

Leontis, N. B. and Westhof, E. (2001). Geometric nomenclature and classification of RNA base pairs. *RNA*, 7(4):499–512.

Leontis, N. B. and Westhof, E. (2003). Analysis of RNA motifs. *Current opinion in structural biology*, 13(3):300–308.

Lescoute, A. and Westhof, E. (2006). Topology of three-way junctions in folded RNAs. *RNA*, 12(1):83–93.

Leutner, M., Gschwind, R. M., Liermann, J., Schwarz, C., Gemmecker, G., and Kessler, H. (1998). Automated backbone assignment of labeled proteins using the threshold accepting algorithm. *Journal of Biomolecular NMR*, 11(1):31–43.

Levin, A., Lis, M., Ponty, Y., O'Donnell, C. W., Devadas, S., Berger, B., and Waldispuhl, Jerome, j. p. y. p. A global sampling approach to designing and reengineering RNA secondary structures.

Li, W. and McMahon, C. A. (2007). A simulated annealing-based optimization approach for integrated process planning and scheduling. *International Journal of Computer Integrated Manufacturing*, 20(1):80–95.

Liang, J. C., Bloom, R. J., and Smolke, C. D. (2011). Engineering biological systems with synthetic RNA molecules. *Molecular cell*, 43(6):915–926.

Lienert, F., Lohmueller, J. J., Garg, A., and Silver, P. A. (2014). Synthetic biology in mammalian cells: next generation research tools and therapeutics. *Nature Reviews Molecular Cell Biology*, 15(2):95–107.

Lilley, D. M. (2004). The Varkud satellite ribozyme. *RNA*, 10(2):151–158.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *The Bell System Technical Journal*, 44(10):2245–2269.

Longfellow, C. E., Kierzek, R., and Turner, D. H. (1990). Thermodynamic and spectroscopic study of bulge loops in oligoribonucleotides. *Biochemistry*, 29(1):278–285.

Lu, H., Setiono, R., and Liu, H. (1996). Effective data mining using neural networks. *IEEE transactions on knowledge and data engineering*, 8(6):957–961.

Lu, Z. J., Gloor, J. W., and Mathews, D. H. (2009). Improved RNA secondary structure prediction by maximizing expected pair accuracy. *RNA*, 15(10):1805–1813.

Lucks, J. B., Qi, L., Mutalik, V. K., Wang, D., and Arkin, A. P. (2011). Versatile RNA-sensing transcriptional regulators for engineering genetic networks. *Proceedings of the National Academy of Sciences*, 108(21):8617–8622.

Luenberger, D. G. (1973). *Introduction to linear and nonlinear programming*, volume 28. Addison-Wesley Reading, MA.

Lyngsø, R. B., Anderson, J. W., Sizikova, E., Badugu, A., Hyland, T., and Hein, J. (2012). Frnakenstein: multiple target inverse RNA folding. *BMC bioinformatics*, 13(1):1.

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

Lysgaard, J., Letchford, A. N., and Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445.

Marimuthu, S., Ponnambalam, S., and Jawahar, N. (2009). Threshold accepting and ant-colony optimization algorithms for scheduling m-machine flow shops with lot streaming. *Journal of materials processing technology*, 209(2):1026–1041.

Marras, A. E., Zhou, L., Su, H.-J., and Castro, C. E. (2015). Programmable motion of DNA origami mechanisms. *Proceedings of the National Academy of Sciences*, 112(3):713–718.

Martinez, H. M. (1984). An RNA folding rule.

Matera, A. G. and Wang, Z. (2014). A day in the life of the spliceosome. *Nature reviews Molecular cell biology*, 15(2):108–121.

Mathews, D. H., Sabina, J., Zuker, M., and Turner, D. H. (1999). Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288(5):911–940.

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

Mattick, J. S. and Makunin, I. V. (2006). Non-coding RNA. *Human molecular genetics*, 15(suppl 1):R17–R29.

McCaskill, J. (1989). The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119.

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

McCown, P. J., Roth, A., and Breaker, R. R. (2011). An expanded collection and refined consensus model of glmS ribozymes. *RNA*, 17(4):728–736.

Méndez-Díaz, I. and Zabala, P. (2006). A branch-and-cut algorithm for graph coloring. *Discrete Applied Mathematics*, 154(5):826–847.

Metropolis, N. and Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341.

Mezmaz, M., Melab, N., and Talbi, E.-G. (2007). A grid-enabled branch and bound algorithm for solving challenging combinatorial optimization problems. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1–9. IEEE.

Miao, Z. and Westhof, E. (2015). Prediction of nucleic acid binding probability in proteins: a neighboring residue network based score. *Nucleic Acids Research*, page gkv446.

Michalewicz, Z. and Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary computation*, 4(1):1–32.

Mitchell, J. E. (2002). Branch-and-cut algorithms for combinatorial optimization problems. *Handbook of applied optimization*, pages 65–77.

Montané, F. A. T. and Galvao, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers and Operations Research*, 33(3):595–619.

Mora-Huertas, C., Fessi, H., and Elaissari, A. (2010). Polymer-based nanocapsules for drug delivery. *International journal of pharmaceutics*, 385(1):113–142.

Mortimer, S. A., Kidwell, M. A., and Doudna, J. A. (2014). Insights into RNA structure and function from genome-wide studies. *Nature reviews Genetics*, 15(7):469–479.

Moscato, P. and Cotta, C. (2002). Memetic algorithms. *Handbook of Applied Optimization*, pages 157–167.

Moscato, P. and Norman, M. G. (1992). A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems. *Parallel computing and transputer applications*, 1:177–186.

Muchmore, S. W., Sattler, M., Liang, H., Meadows, R. P., Harlan, J. E., Yoon, H. S., Nettesheim, D., Chang, B. S., Thompson, C. B., Wong, S.-L., et al. (1996). X-ray and NMR structure of human Bcl-xL, an inhibitor of programmed cell death. *Nature*, 381(6580):335–341.

Mueller, S., Coleman, J. R., Papamichail, D., Ward, C. B., Nimnual, A., Futcher, B., Skiena, S., and Wimmer, E. (2010). Live attenuated influenza virus vaccines by computer-aided rational design. *Nature biotechnology*, 28(7):723–726.

Mulder, S. A. and Wunsch, D. C. (2003). Million city traveling salesman problem solution by divide and conquer clustering with adaptive resonance neural networks. *Neural Networks*, 16(5):827–832.

Nehdi, A., Perreault, J., Beaudoin, J.-D., and Perreault, J.-P. (2007). A novel structural rearrangement of hepatitis delta virus antigenomic ribozyme. *Nucleic acids research*, 35(20):6820–6831.

Nissen, P., Ippolito, J. A., Ban, N., Moore, P. B., and Steitz, T. A. (2001). RNA tertiary interactions in the large ribosomal subunit: the A-minor motif. *Proceedings of the National Academy of Sciences*, 98(9):4899–4903.

Novina, C. D., Murray, M. F., Dykxhoorn, D. M., Beresford, P. J., Riess, J., Lee, S.-K., Collman, R. G., Lieberman, J., Shankar, P., and Sharp, P. A. (2002). siRNA-directed inhibition of HIV-1 infection. *Nature medicine*, 8(7):681–686.

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

Ochoa, G., Hyde, M., Curtois, T., Vazquez-Rodriguez, J., Walker, J., Gendreau, M., Kendall, G., McCollum, B., Parkes, A., Petrovic, S., et al. (2012). Hyflex: A benchmark framework for cross-domain heuristic search. *Evolutionary Computation in Combinatorial Optimization*, pages 136–147.

Oh, I.-S., Lee, J.-S., and Moon, B.-R. (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11):1424–1437.

Oliveira, C. A. and Pardalos, P. M. (2005). A survey of combinatorial optimization problems in multicast routing. *Computers & Operations Research*, 32(8):1953–1981.

Padberg, M. and Rinaldi, G. (1987). Optimization of a 532-city symmetric traveling salesman problem by branch and cut. *Operations Research Letters*, 6(1):1–7.

Padberg, M. and Rinaldi, G. (1991). A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33(1):60–100.

Pan, S.-h. and Malcolm, B. A. (2000). Reduced background expression and improved plasmid stability with pET vectors in BL21 (DE3). *Biotechniques*, 29(6):1234–1238.

Pao, Y. (1989). *Adaptive pattern recognition and neural networks*. Reading, MA (US); Addison-Wesley Publishing Co., Inc.

Parisien, M. and Major, F. (2008). The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature*, 452(7183):51–55.

Pearl, J. (1984). Heuristics: intelligent search strategies for computer problem solving.

Penchovsky, R. and Breaker, R. R. (2005). Computational design and experimental validation of oligonucleotide-sensing allosteric ribozymes. *Nature biotechnology*, 23(11):1424–1433.

Perea, C., Alcala, J., Yepes, V., Gonzalez-Vidosa, F., and Hospitaler, A. (2008). Design of reinforced concrete bridge frames by heuristic optimization. *Advances in Engineering Software*, 39(8):676–688.

Perreault, J., Weinberg, Z., Roth, A., Popescu, O., Chartrand, P., Ferbeyre, G., and Breaker, R. R. (2011). Identification of hammerhead ribozymes in all domains of life reveals novel structural variations. *PLoS Comput Biol*, 7(5):e1002031–e1002031.

Petros, R. A. and DeSimone, J. M. (2010). Strategies in the design of nanoparticles for therapeutic applications. *Nature reviews. Drug discovery*, 9(8):615.

Pley, H. W., Flaherty, K. M., and McKay, D. B. (1994). Hammerhead ribozyme. *Nature*, 372:3.

Pollastri, G., Przybylski, D., Rost, B., and Baldi, P. (2002). Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Bioinformatics*, 47(2):228–235.

Ponty, Y. and Saule, C. (2011). A combinatorial framework for designing (pseudoknotted) RNA algorithms. In *Algorithms in Bioinformatics*, pages 250–269. Springer Berlin Heidelberg.

Powers, T. and Noller, H. F. (1991). A functional pseudoknot in 16S ribosomal RNA. *The EMBO Journal*, 10(8):2203.

Prodhon, C. and Prins, C. (2014). A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17.

Prody, G. A., Bakos, J. T., Buzayan, J. M., Schneider, I. R., and Bruening, G. (1986). Autolytic processing of dimeric plant virus satellite RNA. *Science*, 231:1577–1581.

Prommana, P., Uthaipibull, C., Wongsombat, C., Kamchonwongpaisan, S., Yuthavong, Y., Knuepfer, E., Holder, A. A., and Shaw, P. J. (2013). Inducible knockdown of Plasmodium gene expression using the glmS ribozyme. *PloS One*, 8(8):e73783.

Rahmat-Samii, Y. and Michielssen, E. (1999). Electromagnetic optimization by genetic algorithms. *Microwave Journal*, 42(11):232–232.

Reed, R. and Marks, R. J. (1999). *Neural smithing: supervised learning in feedforward artificial neural networks*. MIT Press.

Reinelt, G. (1994). *The traveling salesman: computational solutions for TSP applications*. Springer-Verlag.

Reinharz, V., Major, F., and Waldispühl, J. (2012). Towards 3D structure prediction of large RNA molecules: an integer programming framework to insert local 3D motifs in RNA secondary structure. *Bioinformatics*, 28(12):i207–i214.

Reinharz, V., Ponty, Y., and Waldispühl, J. (2013). A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution. *Bioinformatics*, 29(13):i308–i315.

Ren, J., Rastegari, B., Condon, A., and Hoos, H. H. (2005). HotKnots: Heuristic prediction of RNA secondary structures including pseudoknots. *RNA*, 11(10):1494–1504.

Retwitzer, M. D., Kifer, I., Sengupta, S., Yakhini, Z., and Barash, D. (2015). An efficient minimum free energy structure-based search method for riboswitch identification based on inverse RNA folding. *PloS one*, 10(7):e0134262.

Reuter, J. S. and Mathews, D. H. (2010). RNAstructure: software for RNA secondary structure prediction and analysis. *BMC bioinformatics*, 11(1):129.

Reynolds, A., Leake, D., Boese, Q., Scaringe, S., Marshall, W. S., and Khvorova, A. (2004). Rational siRNA design for RNA interference. *Nature Biotechnology*, 22(3):326–330.

Richardson, J. S. (1981). The anatomy and taxonomy of protein structure. *Advances in Protein Chemistry*, 34:167–339.

Richardson, J. S., Schneider, B., Murray, L. W., Kapral, G. J., Immormino, R. M., Headd, J. J., Richardson, D. C., Ham, D., Hershkovits, E., Williams, L. D., et al. (2008). RNA backbone: consensus all-angle conformers and modular string nomenclature (an RNA Ontology Consortium contribution). *RNA*, 14(3):465–481.

Richer, J.-M., Goëffon, A., and Hao, J.-K. (2009). A memetic algorithm for phylogenetic reconstruction with maximum parsimony. *EvoBIO*, 9:164–175.

Rivas, E. and Eddy, S. R. (1999). A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology*, 285(5):2053–2068.

Rodrigo, G., Landrain, T. E., Shen, S., and Jaramillo, A. (2013). A new frontier in synthetic biology: automated design of small RNA devices in bacteria. *Trends in Genetics*, 29(9):529–536.

Rose, P. W., Bi, C., Bluhm, W. F., Christie, C. H., Dimitropoulos, D., Dutta, S., Green, R. K., Goodsell, D. S., Prlić, A., Quesada, M., et al. (2013). The RCSB protein data bank: new resources for research and education. *Nucleic Acids Research*, 41(D1):D475–D482.

Rossi, R. A., Gleich, D. F., Gebremedhin, A. H., and Patwary, M. M. A. (2014). Fast maximum clique algorithms for large graphs. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 365–366. ACM.

Rost, B. and Sander, C. (1994). Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins: Structure, Function, and Bioinformatics*, 19(1):55–72.

Roth, A., Weinberg, Z., Chen, A. G., Kim, P. B., Ames, T. D., and Breaker, R. R. (2014). A widespread self-cleaving ribozyme class is revealed by bioinformatics. *Nature chemical biology*, 10(1):56–60.

Roussel, O. and Soria, M. (2009). Boltzmann sampling of ordered structures. *Electronic Notes in Discrete Mathematics*, 35:305–310.

Roy, H. and Ibba, M. (2006). Molecular biology: sticky end in protein synthesis. *Nature*, 443(7107):41–42.

Rubinstein, R. Y. and Kroese, D. P. (2016). *Simulation and the Monte Carlo method*, volume 10. John Wiley and Sons.

Ruder, W. C., Lu, T., and Collins, J. J. (2011). Synthetic biology moving into the clinic. *Science*, 333(6047):1248–1252.

Rudolph, G. (1994). Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5(1):96–101.

Ruzzo, W. L. and Gorodkin, J. (2014). *De novo* discovery of structured ncRNA motifs in genomic sequences. *RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods*, pages 303–318.

Sankoff, D. (1985). Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM Journal on Applied Mathematics*, 45(5):810–825.

Sato, K., Kato, Y., Hamada, M., Akutsu, T., and Asai, K. (2011). IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. *Bioinformatics*, 27(13):i85–i93.

Scarborough, R. J., Lévesque, M. V., Perreault, J.-P., and Gatignol, A. (2014). Design and evaluation of clinically relevant SOFA-HDV ribozymes targeting HIV RNA. *Therapeutic Applications of Ribozymes and Riboswitches: Methods and Protocols*, pages 31–43.

Schmitz, M. and Steger, G. (1996). Description of RNA folding by "simulated annealing". *Journal of Molecular Biology*, 255(1):254–266.

Schnall-Levin, M., Chindelevitch, L., and Berger, B. (2008). Inverting the Viterbi algorithm: an abstract framework for structure design. In *Proceedings of the 25th international conference on Machine learning*, pages 904–911. ACM.

Schultes, E. A. and Bartel, D. P. (2000). One sequence, two ribozymes: implications for the emergence of new ribozyme folds. *Science*, 289(5478):448–452.

Schwefel, H.-P. (1975). *Evolutionsstrategie und numerische Optimierung*. PhD thesis, Technische Universitat Berlin.

Seeman, N. C. (2010). Nanomaterials based on DNA. *Annual Review of Biochemistry*, 79:65–73.

Serganov, A. and Nudler, E. (2013). A decade of riboswitches. *Cell*, 152(1):17–24.

Serra, M. J. and Turner, D. H. (1995). Predicting thermodynamic properties of RNA. *Methods in Enzymology*, 259:242–261.

Shapiro, B. A., Bindewald, E., Kasprzak, W., and Yingling, Y. (2008). Protocols for the in silico design of RNA nanostructures. *Nanostructure Design: Methods and Protocols*, pages 93–115.

Shapiro, B. A., Yingling, Y. G., Kasprzak, W., and Bindewald, E. (2007). Bridging the gap in RNA structure prediction. *Current opinion in structural biology*, 17(2):157–165.

Shen, Y. and Bax, A. (2013). Protein backbone and sidechain torsion angles predicted from NMR chemical shifts using artificial neural networks. *Journal of Biomolecular NMR*, 56(3):227–241.

Sheng, Q., Moreau, Y., and De Moor, B. (2003). Biclustering microarray data by Gibbs sampling. *Bioinformatics*, 19(suppl_2):ii196–ii205.

Shintani, K., Imai, A., Nishimura, E., and Papadimitriou, S. (2007). The container shipping network design problem with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 43(1):39–59.

Shmygelska, A. and Hoos, H. H. (2005). An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. *BMC bioinformatics*, 6(1):30.

Shu, D., Li, H., Shu, Y., Xiong, G., Carson III, W. E., Haque, F., Xu, R., and Guo, P. (2015). Systemic delivery of anti-miRNA for suppression of triple negative breast cancer utilizing RNA nanotechnology. *Acs Nano*, 9(10):9731.

Shum, K.-T. and Rossi, J. J. (2013). RNA Nanotechnology approach for targeted delivery of RNA therapeutics using cell-internalizing aptamers. In *DNA and RNA Nanobiotechnologies in Medicine: Diagnosis and Treatment of Diseases*, pages 395–423. Springer Berlin Heidelberg.

Sietsma, J. and Dow, R. J. (1991). Creating artificial neural networks that generalize. *Neural networks*, 4(1):67–79.

Singer, M. F. and Leder, P. (1966). Messenger RNA: an evaluation. *Annual Review of Biochemistry*, 35(1):195–230.

Smith, A. M., Fuchs, R. T., Grundy, F. J., and Henkin, T. (2010). Riboswitch RNAs: regulation of gene expression by direct monitoring of a physiological signal. *RNA biology*, 7(1):104–110.

Smith, K. A. (1999). Neural networks for combinatorial optimization: a review of more than a decade of research. *INFORMS Journal on Computing*, 11(1):15–34.

Song, Y., Zhang, C., and Fang, Y. (2008). Multiple multidimensional knapsack problem and its applications in cognitive radio networks. In *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pages 1–7. IEEE.

Sörensen, K. and Sevaux, M. (2006). MA— PM: memetic algorithms with population management. *Computers & Operations Research*, 33(5):1214–1225.

Soukup, G. A. (2006). Core requirements for glmS ribozyme self-cleavage reveal a putative pseudoknot structure. *Nucleic Acids Research*, 34(3):968–975.

Spieth, C., Streichert, F., Speer, N., and Zell, A. (2004). A memetic inference method for gene regulatory networks based on s-systems. volume 1, pages 152–157. IEEE.

Staple, D. W. and Butcher, S. E. (2005). Pseudoknots: RNA structures with diverse functions. *PLoS Biol*, 3(6):e213.

Stark, A., Lin, M. F., Kheradpour, P., Pedersen, J. S., Parts, L., Carlson, J. W., Crosby, M. A., Rasmussen, M. D., Roy, S., Deoras, A. N., et al. (2007). Discovery of functional elements in 12 Drosophila genomes using evolutionary signatures. *Nature*, 450(7167):219–232.

Stefani, G. and Slack, F. J. (2008). Small non-coding RNAs in animal development. *Nature Reviews Molecular Cell Biology*, 9(3):219–230.

Stombaugh, J., Zirbel, C. L., Westhof, E., and Leontis, N. B. (2009). Frequency and isostericity of RNA base pairs. *Nucleic Acids Research*, 37(7):2294–2312.

Stützle, T. and Hoos, H. H. (2000). Max–min ant system. *Future generation computer systems*, 16(8):889–914.

Sun, W., Jiang, T., Lu, Y., Reiff, M., Mo, R., and Gu, Z. (2014). Cocoon-like self-degradable DNA nanoclew for anticancer drug delivery. *Journal of the American Chemical Society*, 136(42):14722–14725.

Svozil, D., Kvasnicka, V., and Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62.

Takefuji, Y. and Lee, K. C. (1991). Artificial neural networks for four-coloring map problems and K-colorability problems. *IEEE Transactions on Circuits and Systems*, 38(3):326–333.

Taneda, A. (2011). MODENA: a multi-objective RNA inverse folding. *Adv. Appl. Bioinform. Chem*, 4:1–12.

Taneda, A. (2012). Multi-objective genetic algorithm for pseudoknotted RNA sequence design. *Frontiers in Genetics*, 3:36.

Taneda, A. (2015). Multi-objective optimization for RNA design with multiple target secondary structures. *BMC Bioinformatics*, 16(1):280.

Tang, H. and Miller-Hooks, E. (2005). A tabu search heuristic for the team orienteering problem. *Computers and Operations Research*, 32(6):1379–1407.

Theis, C., Janssen, S., and Giegerich, R. (2010). Prediction of RNA secondary structure including kissing hairpin motifs. In *International Workshop on Algorithms in Bioinformatics*, pages 52–64. Springer.

Thijs, G., Marchal, K., Lescot, M., Rombauts, S., De Moor, B., Rouzé, P., and Moreau, Y. (2002). A gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *Journal of Computational Biology*, 9(2):447–464.

Treiber, D. K. and Williamson, J. R. (2001). Beyond kinetic traps in RNA folding. *Current Opinion in Structural Biology*, 11(3):309–314.

Tsai, M.-C., Manor, O., Wan, Y., Mosammaparast, N., Wang, J. K., Lan, F., Shi, Y., Segal, E., and Chang, H. Y. (2010). Long noncoding RNA as modular scaffold of histone modification complexes. *Science*, 329(5992):689–693.

Van Batenburg, F., Gultyaev, A. P., Pleij, C., Ng, J., and Oliehoek, J. (2000). Pseudobase: a database with RNA pseudoknots. *Nucleic Acids Research*, 28(1):201–204.

Van Laarhoven, P. J. and Aarts, E. H. (1987). Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer.

Varani, G. and Tinoco, I. (1991). RNA structure and NMR spectroscopy. *Quarterly reviews of biophysics*, 24(04):479–532.

Visée, M., Teghem, J., Pirlot, M., and Ulungu, E. L. (1998). Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem. *Journal of Global Optimization*, 12(2):139–155.

Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al. (2001). Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584.

Wahlestedt, C. (2013). Targeting long non-coding RNA to therapeutically upregulate gene expression. *Nature Reviews Drug Discovery*, 12(6):433–446.

Waldispühl, J., Devadas, S., Berger, B., and Clote, P. (2008). Efficient algorithms for probing the RNA mutation landscape. *PLoS Comput Biol*, 4(8):e1000124.

Waxman, B. M. (1988). Routing of multipoint connections. *IEEE journal on selected areas in communications*, 6(9):1617–1622.

Weber, F., Wagner, V., Rasmussen, S. B., Hartmann, R., and Paludan, S. R. (2006). Double-stranded RNA is produced by positive-strand RNA viruses and DNA viruses but not in detectable amounts by negative-strand RNA viruses. *Journal of Virology*, 80(10):5059–5064.

Weinbrand, L., Avihoo, A., and Barash, D. (2013). RNAfbinv: an interactive java application for fragment-based design of RNA sequences. *Bioinformatics*, page btt494.

Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Winfree, E., Liu, F., Wenzler, L. A., and Seeman, N. C. (1998). Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394(6693):539–544.

Winkler, W. C., Nahvi, A., Roth, A., Collins, J. A., and Breaker, R. R. (2004). Control of gene expression by a natural metabolite-responsive ribozyme. *Nature*, 428(6980):281–286.

Winston, W. L. and Goldberg, J. B. (2004). *Operations research: applications and algorithms*, volume 3. Thomson Brooks/Cole Belmont.

Witwer, C., Hofacker, I. L., and Stadler, P. F. (2004). Prediction of consensus RNA secondary structures including pseudoknots. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(2):66–77.

Woeginger, G. J. (2003). Exact algorithms for NP-hard problems: A survey. *Lecture notes in computer science*, 2570(2003):185–207.

Wolfe, B. R. and Pierce, N. A. (2014). Sequence design for a test tube of interacting nucleic acid strands. *ACS synthetic biology*, 4(10):1086–1100.

Wu, X., Zhu, X., Wu, G.-Q., and Ding, W. (2014). Data mining with big data. *IEEE transactions on knowledge and data engineering*, 26(1):97–107.

Wyatt, J. R., Puglisi, J. D., and Tinoco, I. (1989). RNA folding: pseudoknots, loops and bulges. *BioEssays*, 11(4):100–106.

Xie, Z., Wroblewska, L., Prochazka, L., Weiss, R., and Benenson, Y. (2011). Multi-input RNAi-based logic circuit for identification of specific cancer cells. *Science*, 333(6047):1307–1311.

Yang, H., McLaughlin, C. K., Aldaye, F. A., Hamblin, G. D., Rys, A. Z., Rouiller, I., and Sleiman, H. F. (2009). Metal–nucleic acid cages. *Nature Chemistry*, 1(5):390–396.

Yang, Y. (2015). Introduction: Overview of DNA Origami as Biomaterials and Application. In *Artificially Controllable Nanodevices Constructed by DNA Origami Technology*, pages 1–19. Springer.

Yu, Z., Jinhai, L., Guochang, G., Rubo, Z., and Haiyan, Y. (2002). An implementation of evolutionary computation for path planning of cooperative mobile robots. In *Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on*, volume 3, pages 1798–1802. IEEE.

Zadeh, J. N., Steenberg, C. D., Bois, J. S., Wolfe, B. R., Pierce, M. B., Khan, A. R., Dirks, R. M., and Pierce, N. A. (2011a). NUPACK: analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, 32(1):170–173.

Zadeh, J. N., Wolfe, B. R., and Pierce, N. A. (2011b). Nucleic acid sequence design via efficient ensemble defect optimization. *Journal of Computational Chemistry*, 32(3):439–452.

Zalatan, J. G., Lee, M. E., Almeida, R., Gilbert, L. A., Whitehead, E. H., La Russa, M., Tsai, J. C., Weissman, J. S., Dueber, J. E., Qi, L. S., et al. (2015). Engineering complex synthetic transcriptional programs with CRISPR RNA scaffolds. *Cell*, 160(1):339–350.

Zandi, K., Butler, G., and Kharma, N. (2016). An adaptive defect weighted sampling algorithm to design pseudoknotted RNA secondary structures. *Frontiers in Genetics*, 7:129.

Zandi, K., Najeh, S., Kharma, N., and Perreault, J. (2018). Computational design and experimental validation of pseudoknotted ribozymes. *RNA*.

Zhou, J., Satheesan, S., Li, H., Weinberg, M. S., Morris, K. V., Burnett, J. C., and Rossi, J. J. (2015). Cell-specific RNA aptamer against human CCR5 specifically targets HIV-1 susceptible cells and inhibits HIV-1 infectivity. *Chemistry & biology*, 22(3):379–390.

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

Zuker, M. and Sankoff, D. (1984). RNA secondary structures and their prediction. *Bulletin of mathematical biology*, 46(4):591–621.

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

# Appendix A

# Designed sequences for mouse gut metagenome

The following table presents the dataset for the mouse gut metagenome generated by Enzymer in Chapter 5. Bold letters represent the catalytic core of the ribozyme as inferred by MSA analysis

| Annotation | Designed RNA sequences |
|---|---|
| $\phi^{1}_{HH}$ | CCGUCGCAAA AAGGGU**CCUG AUGAG**CAAGC GACAAAAAAA **GCGAAA**CACC GCGAAAAAGC GGUG**UCG**ACC CGAGAAAAAA G |
| $\phi^{2}_{HH}$ | CUGAUAGACC CCGGAU**CCUG AUGAG**CUACU AUCCCUAAAU **GCGAAA**CACA GGCAUGAACC UGUG**UCG**AUC CUAUAAAACC C |
| $\phi^{3}_{HH}$ | CCCCUCUAAA AAGGGA**CCUG AUGAG**CCCAG AGGAAAAAAC **GCGAAA**GGCU GCUAAAGUGU AGUC**UCG**UCC CAAACAACAU A |
| $\phi^{4}_{HH}$ | CAGUUCGAAA AAGCCU**CCUG AUGAG**CAACG AACAAACCUA **GCGAAA**CCGU GGUUAACUCC AUGG**UCG**AGG CGACAAAAAA U |
| $\phi^{5}_{HH}$ | AACGGAGCCC UUCCGC**CCUG AUGAG**CAACU CUGAAUAAAA **GCGAAA**CUGU AGAACUACCU ACGG**UCGGCG** GUUUUCUAUA C |
| $\phi^{6}_{HH}$ | CCCCUCGAAA AAGUGU**CCUG AUGAG**CAACG AGGAACCCCC **GCGAAA**GGCG UGAAAUACCG CGUC**UCG**ACA CAAGAGAAAA G |
| $\phi^{7}_{HH}$ | ACGGACCACC CCCAGU**CCUG AUGAG**CAAGG UCCAAAAAAA **GCGAAA**CUUG AUGUAAUAGU UAAG**UCG**ACU GAAAAAAACC A |
| $\phi^{8}_{HH}$ | ACGGAGGGUG UGGGGC**CCUG AUGAG**CUGCC UCCUUGAAUU **GCGAAA**GUUG GAAUGAUCUC UAAC**UCG**GCC CGUUGAGUUG U |
| $\phi^{wild}_{HH}$ | GGUACCGAAU AAAUCC**CCUG AUGAG**CAACG GUGAGAGCCG **GCGAAA**CUAC CCAAACAAGG GUAG**UCG**GGA UAGUACCAUA A |
| Design template $t_{HH}$ | oooooooooo oooooo**CCUG AUGAG**ooooo oooooooooo **GCGAAA**oooo oooooooooo oooo**UCG**ooo oooooooooo o |
| Secondary structure | ..[[[[[.....(((((......(((..]]]]].......)))..(((((((......))))))).))))).......... |

Table 5: Designed sequences for the mouse gut metagenome hammerhead ribozyme.

# Appendix B

# *in-vitro* transcription protocols

## B.1 Transcription template preparation

This section presents the wet-lab protocols used to transcribe the ribozyme sequences for *in-vitro* experiments described in Chapter 6.

The PCR reaction was made with 100 $\mu$M dNTP, 1X thermopol Taq reaction buffer (20 mM Tris-HCl, pH 8.8, 10 mM $(NH_4)2SO_4$, 10 mM KCl, 2 mM $MgSO_4$, 0.1% Triton X-100), 1X of Q solution (Qiagen) and 1 $\mu$M of each primer at their corresponding annealing temperatures. The 14 sequences generated for the mouse gut metagenome hammerhead were amplified by a PCR of 15 cycles. 1 $\mu$M of a primer with the T7 promoter sequence was used because each ribozyme oligonucleotide sequence already has a T7 promoter, required for transcription with the T7 RNA polymerase. The oligonucleotides to generate the DNA templates for ribozymes were ordered from BioCorp DNA Inc, IDT or sigma.

The wild-type ribozyme (from the mouse gut metagenome) was used as a positive control. We made five mutations in the catalytic core of one of the ribozymes generated by Enzymer (in MHHRz3) to use as a size marker for uncleaved RNA. For the *Yarrowia lypolityca* HHRz, the two designed sequences and the wild type sequence were amplified by a PCR of 25 cycles. Inactive strains of each ribozyme were also amplified to be used as uncleaved size markers. We also amplified the WT glmS ribozyme, the two designed sequences and their modified strains with a PCR of 30 cycles. To observe the results of the cleavage reaction of the glmS ribozyme that naturally generates one nucleotide at the $5'$ side we added a sequence of 20 adenosines (As) at the $5'$ end.

## B.2 Transcription

The synthesis of the ribozymes by *in-vitro* transcription was made in the presence of radioactive UTP32 to be able to visualize and quantify the self-cleavage activity during the transcription. The reaction of 50 $\mu$l contained the DNA produced by PCR, 2 mM rNTPS (2 mM each of ATP, GTP, CTP) and 0.8 mM UTP, 1X transcription buffer (80 mM HEPES-KOH pH 7.5, 24 mM MgCl2, 40 mM DTT, 2 mM spermidine), 1U/$\mu$l of inorganic pyrophosphatase (Sigma Aldrich, USA) , 1U/$\mu$l RNase inhibitor (Thermofisher), 1U/$\mu$l T7 RNA polymerase and milliQ water for 2 hours at 37°C. Then 1U/$\mu$l of Dnase (RNase free) (NEB) was added to each reaction with incubation for 30 min at 37°C. For the glmS ribozymes, for each ribozyme we had two reactions with or without 1 mM of GlcN6P.

## B.3 Self-cleavage analysis

An aliquot of 1 $\mu$l of each ribozyme transcription product was taken and diluted in 9 $\mu$l of milliQ water. To each aliquot an equal volume of the 2X dye formamide buffer (95% formamide, 10 mM EDTA, 0.025% bromophenol blue and 0.025% xylene cyanol blue) was added. The full length and the cleaved ribozymes were separated by electrophoresis on a 1% polyacrylamide gel in 1X TBE buffer (89 mM Tris, 89 mM boric acid, 0.02M EDTA). The gel was exposed to a storage phosphor screen for (30 min to 1 hour) and scanned with a Typhoon FLA9500 (GE Health Care) and quantified with ImageQuant.

# Appendix C

# *in-vivo* transcription protocols

This section presents the plasmid and *in-vivo* protocols used in Chapter 6.

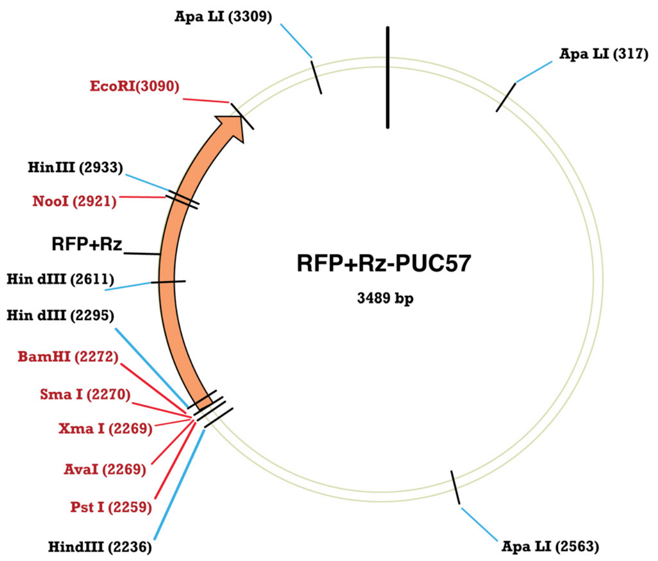


Figure 32: PUC57 plasmid

Plasmids (pUC57) containing the two active or mutated ribozymes were each digested by SacI and SphI for 1 hour at 37°C in the presence of 1 x cutsmart buffer. Digestion products were purified on a 1% agarose gel to extract the insert to be recloned, the band corresponding to the ribozyme-RFP insert was cut and the DNA extracted by the EZ spin column plasmid DNA kit Kit (Biobasic).

Ligation of the plasmid digested with the insert (at a 1:10 ratio) was made at room temperature for two hours in the presence of T4 ligase and its 1X buffer (50 mM Tris-HCl, 10 mM MgCl2, 1 mM ATP, 10 mM DTT, pH 7.5). Subsequently, the plasmids were transformed into E. coli BL21 DE3 competent cells.

The next day three colonies of each culture were stitched and seeded in LB in the presence of ampicillin and incubated overnight at 37°C under rotation. The next day, the OD of each culture was measured by the nanodrop, a triplicate for each culture was used to make a technical triplicate with three different concentrations of IPTG (1mM, 2mM, 4mM). From each culture 1 ml was diluted in 9 ml of LB to eventually have 36 cultures (three cultures each at a concentration of IPTG from each of 12 cultures initiated from the initial stitched colonies). The cultures were incubated at 37°C and OD and fluorescence intensity were measured (using typhoon FLA) once every hour for five hours and after overnight culture.