From RNA folding to inverse folding:
a computational study

Der Fakultät für Mathematik und Informatik
der Universität Leipzig
eingereichte

D I S S E R T A T I O N

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr.rer.nat.)

im Fachgebiet

Informatik

vorgelegt

von Diplominformatiker **Nono Saha Cyrille Merleau**
geboren am 26-03-1992 in Bafoussam, Kamerun

Leipzig, den December 2021

Ohana means family.
Family means nobody gets left behind, or forgotten.
— Lilo & Stitch

Dedicated to my loving dad Michel Saha.

ABSTRACT

Since the discovery of the structure of Deoxyribonucleic acid (DNA) in the early 1953s, and its double-chained complement of information hinting at its means of replication, biologists have recognized the strong connection between molecular structure and function. In the past two decades, there has been a surge of research on an ever-growing class of Ribonucleic acid (RNA) molecules that are non-coding but whose various folded structures allow a diverse array of vital functions. From the well-known splicing and modification of ribosomal RNA, non-coding RNAs (ncRNAs) are now known to be intimately involved in possibly every stage of DNA translation and protein transcription, as well as RNA signalling and gene regulation processes.

Despite the rapid development and declining cost of modern molecular methods, they typically can only describe ncRNA's structural conformations *in vitro*, which differ from their *in vivo* counterparts. Moreover, it is estimated that only a tiny fraction of known ncRNA has been documented experimentally, often at a high cost. There is thus a growing realization that computational methods must play a central role in the analysis of ncRNAs. Not only do computational approaches hold the promise of rapidly characterizing many ncRNAs yet to be described, but there is also the hope that by understanding the rules that determine their structure, we will gain better insight into their function and design. Many studies revealed that the ncRNA functions are performed by high-level structures that often depend on their low-level structures, such as the secondary structure. This thesis studies the computational folding mechanism and inverse folding of ncRNAs at the secondary level.

In this thesis, we describe the development of two bioinformatic tools that have the potential to improve our understanding of RNA secondary structure. These tools are as follows: (1) `RAFFT` for efficient prediction of pseudoknot-free RNA folding pathways using the fast Fourier transform (FFT); (2) `aRNAque`, an evolutionary algorithm inspired by Lévy flights for RNA inverse folding with or without pseudoknot (A secondary structure that often poses difficulties for bio-computational detection).

The first tool, `RAFFT`, implements a novel heuristic to predict RNA secondary structure formation pathways that has two com-

ponents: (i) a folding algorithm and (ii) a kinetic ansatz. When considering the best prediction in the ensemble of 50 secondary structures predicted by RAFFT, its performance matches the recent deep-learning-based structure prediction methods. RAFFT also acts as a folding kinetic ansatz, which we tested on two RNAs: the coronavirus frameshifting stimulation element (CFSE) and a classic bi-stable sequence. In both test cases, fewer structures were required to reproduce the full kinetics, whereas known methods (such as Treekin) required a sample of 20,000 structures and more.

The second tool, aRNAque, implements an evolutionary algorithm (EA) inspired by the Lévy flight, allowing both local global search, and which supports pseudoknotted target structures. The number of point mutations at every step of aRNAque EA is drawn from a Zipf distribution. Therefore, our proposed method increases the diversity of designed RNA sequences and reduces the average number of evaluations of the evolutionary algorithm. The overall performance showed improved empirical results compared to existing tools through intensive benchmarks on both pseudoknotted and pseudoknot-free datasets.

In conclusion, we highlight some promising extensions of the versatile RAFFT's method to RNA-RNA interaction studies. We also provide an outlook of both tools' implications in studying evolutionary dynamics.

PUBLICATIONS

This thesis presents our contributions to RNA secondary structures' computational methods for the folding and inverse folding problems. They were obtained in collaboration with my advisor Matteo Smerlak, Vaitea Opuu and Vincent Messow. Most of the ideas and figures have appeared previously in the following publications:

- [128] **Nono SC Merleau** and Matteo Smerlak (2021). *A simple evolutionary algorithm guided by local mutations for an efficient RNA design*. In: *Proceedings of the Genetic and Evolutionary Computation Conference.* pp. 1027-1034.

- [138] Vaitea Opuu, **Nono SC Merleau**, Vincent Messow, and Matteo Smerlak(2021). *RAFFT: Efficient prediction of RNA folding pathways using the fast Fourier transform*. In: *bioRxiv* (Submitted to PLoS Comput. Biol.)

- [129] **Nono SC Merleau** and Matteo Smerlak (2022). *An evolutionary algorithm for inverse RNA folding inspired by Lévy flights*. In: *bioRxiv* (Submited and Accepted) (At BMC Bioinformatics).

In addition to these works in RNA folding and inverse folding, I studied the fragility of RNA viruses during my PhD using multi-agent evolutionary algorithm simulations. I also contributed to various works in natural language processing and multi-agent simulations for Holonification models. None of these investigations,

- Igor Haman Tchappi, Stéphane Galland, Vivient Corneille Kamla, Jean-Claude Kamgang, **Cyrille Merleau S Nono**, and Hui Zhao (2019). *Holonification model for a multilevel agent-based system*. In: *Personal and Ubiquitous Computing* 23(5).

- Ivan P Yamshchikov, **Cyrille Merleau Nono Saha**, Igor Samenko, Jürgen Jost (2020). *It Means More if It Sounds Good: Yet Another Hypothesis Concerning the Evolution of Polysemous Words*. In: *Proceedings of the 5th International Conference on Complexity, Future Information Systems and Risk (COMPLEXIS 2020)*, pages 143-148.

- **Nono SC Merleau**, Sophie Pénisson, Philip J Gerrish, Santiago F Elena, and Matteo Smerlak (2021). *Why are viral genomes so fragile? The bottleneck hypothesis*. In: *PLoS. Comput Biol*. 17(7).

will be addressed in this manuscript.

We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.

— Donald E. Knuth [104]

ACKNOWLEDGEMENTS

Acknowledgements to be put here.

Many thanks to everybody who is already reading through this first draft!

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

LIST OF SYMBOLS AND ACRONYMS

DNA Deoxyribonucleic acid

RNA Ribonucleic acid

ncRNA non-coding RNA

CFSE coronavirus frameshifting stimulation element

EA evolutionary algorithm

lncRNA long non-coding RNA

sncRNA short non-coding RNA

tRNA transfert RNA

rRNA ribosomal RNA

cRNA coding RNA

mRNA messenger RNA

CRISPR clustered regularly interspaced short palindromic repeats

SELEX systematic evolution of ligands by exponential enrichment

MFE minimum free energy

DP dynamic programming

NMR nuclear magnetic resonance

ML machine learning

DNN deep neural network

SCFG stochastic context-free grammar

SVM support vector machine

CLLM conditional log-linear model

NN nearest neighbour

MEA maximum expected accuracy

GPU graphics processing unit

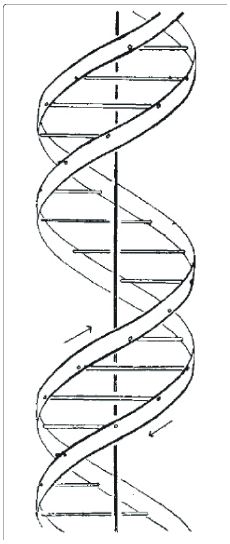
WC Watson-Crick

PCA principal components analysis
PC principal component
PPV positive predictive value
API application programming interface
FFT fast Fourier transform
SAVE synthetic attenuated virus engineering
NP non-deterministic polynomial-time
CPU central processing unit
NMCS nested monte carlo search
MCTS monte carlo tree search
ED ensemble defect
NED normalyzed energy distance
MPGA massively parallel genetic algorithm
piRNA PIWI-interacting RNA
PAR promoter-associated RNA
miRNA microRNA
snoRNA small nucleolar RNA
DFT distcrete Fourier transform
IDFT Inverse discrete Fourier transform
NMR nuclear magnetic resonance

1 INTRODUCTION

1.1 SURVEY

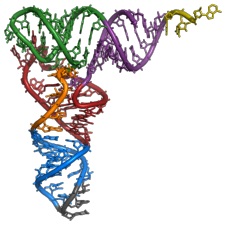
DNAs and RNAs are macromolecules in the nucleus of eukaryotic cells that allow storing information with the help of nucleotides. Nucleotides consist of a five-carbon sugar, a phosphate group, and a nucleobase. There are four nucleotides in the DNA, distinguished by their nucleobases: A for Adenine, T for Thymine, G for Guanine, and C for Cytosine. Similar to DNA, we also find four different nucleotides in RNA, also distinguished by their nucleobases with only one exception; the Uracil (U), which replaces Thymine in DNA. Even though the basis blocks constituting the DNA were known for many years, in 1953, James Watson and Francis Crick [208] succeeded in putting them together and suggested a reasonable DNA structure. Their work revealed for the first time that the structure of DNA molecules has helical chains, each coiled round the same axis where the chain consists of phosphate dieter groups. The two chains are held together by the purines and pyrimidine bases; they are joined together in pairs, a single base from the other chain bonded to a single base from one chain. For the biding to occur, one of the pairs must be Adenine and Thymine or Guanine and Cytosine. A DNA molecule structure is depicted on the right side of the page. In contrast to DNA, RNAs are mostly single-stranded, and the complementary pairings formed in the structure are A-U, G-U and G-C.



Helical representation of DNA structures [208].

Watson and Crick's elucidation of DNA structure has motivated many other scientists to investigate further the structural implications of molecules in functions such as replication and gave rise to modern molecular biology. Later in the same year, Crick formulated the central dogma of molecular biology that describes the flow of information from DNA to messenger RNA (mRNA) through transcription and from mRNAs to proteins through translation [26]. Since this information flow was proposed, more works have been done to investigate each step.

But not all RNAs are translated into proteins; in other terms, not all RNAs are mRNAs. There are mainly two RNA groups: coding RNAs (cRNAs) that are translated into proteins, and non-coding RNAs that are not translated into proteins. During the transcrip-



The tertiary structure of tRNA. The CCA-tail is in yellow, the acceptor stem in purple, the variable loop in orange, D-arm in red, the anticodon arm in blue with anticodon in black, and T-arm in green (Taken from Wikipedia)

tion and translation steps in the information flow, some vital functions are performed by ncRNAs such as ribosomal RNA (rRNA) and tRNA. The tertiary structure of a tRNA is shown on the left side of the page. The study of such RNAs revealed that rRNAs, rather than ribosomal proteins, catalyze the synthesis of proteins (i.e. the polymerization of amino acids), distinguish between correct and incorrect codon-anticodon pairs and prevent the premature hydrolysis of peptidyl-tRNAs [15, 133]. Apart from being central to the protein machinery, ncRNAs regulate various biological functions in transcriptional interference, telomere maintenance, epigenetic changes, imprinting, post-transcriptional, translational control, structural organization, cell differentiation and development [51, 159]. We are interested in this work in the structures of ncRNAs.

The function of ncRNAs is largely determined by their high-dimensional structure [19]. For instance, we can analyze the catalytic function of ribozymes in terms of basic structural motifs, e.g. hammerhead or hairpin structures [39]. Other RNAs, like riboswitches, involve changes between alternative structures [202]. Understanding the relation sequence and structure is a central challenge in molecular biology. In the last 20 years, many different methods for determining the RNA structures of molecules have emerged: from experimental lab methods to computational approaches. For experimental lab methods, X-ray crystallography and the nuclear magnetic resonance (NMR) are the most accurate approaches to offer structural information at a single base-pair resolution. Both experimental methods are often characterized by high experimental cost and low throughput. In addition to those limitations, RNA molecules are volatile and difficult to crystallize.

Despite the development of more sophisticated techniques to infer the state of nucleotides in RNA molecules using enzymatic [95, 200] or chemical probes [193, 212] coupled with next-generation sequencing [12, 192], most of them can only capture RNA structures *in vitro* which mostly differ from the *in vivo* structure conformations. Experimentally, only a tiny fraction of known ncRNAs has been determined [142]. Because measuring the structure of RNAs experimentally is very difficult and expensive, computational approaches play a central role in the analysis of natural RNAs [50, 167], and are an essential alternative to experimental approaches.

Given the ncRNA sequence of bases (primary structure), RNAs fold into secondary structures, such as stem loops and pseudoknots, before folding into higher level (tertiary and quaternary)

structures [16, 194]. This separation of time scales justifies focusing on the secondary structure prediction; evidence suggests that the RNA's secondary structures largely determine the resulting high-level structures [194].

This thesis focuses on computational methods addressing RNA molecules' folding and inverse folding at the secondary level. This introductory chapter presents a brief overview of the non-coding RNA concepts. The overview concepts contain biological and biochemical structure definitions of the non-coding RNAs. It also gives an overview of different techniques used to identify new ncRNAs and some applications. It concludes by providing the bioinformatic definitions of RNA secondary structure that constitute the basis and understanding of computational methods and the results presented in this thesis.

1.2 CHARACTERISTICS AND BIOLOGICAL FUNCTIONS OF NCRNA

In the previous section, we introduced the classical view of information flow in microbiology. Two important ncRNAs involved in the protein machinery have been highlighted (tRNA and rRNA). In this section, we provide some of the main characteristics of ncRNAs, and we emphasize how those characteristics often play an essential role in realizing their functions.

What motivates the computational studies of ncRNAs is often the importance of the biological function they play. Consequently, the ncRNAs can be classified based on their biological functions. Although many recent transcriptomic and bioinformatic studies suggested thousands of ncRNAs with their functional importances, the total number of ncRNAs encoded in the human genome still remains unknown [159]. More recently, newly identified ncRNAs have not been validated by their function; it could be possible that most of them are non-functional. Some evolutionary experiments *in vitro* have shown that RNA molecules can catalyze various chemical reactions relevant to biological processes such as RNA replication, nucleotide synthesis, thymidylate synthesis, lipid synthesis, and sugar metabolism [45, 153]. Another characteristic of ncRNAs is their lengths formed post-transcriptionally. We often distinguish two main ncRNA classes of critical biological functions: the short non-coding RNAs (sncRNAs) (with length $< 30nt$) and the long non-coding RNAs (lncRNAs) (with length $> 200nt$). The length limit is often because of the practical considerations, including separating RNAs in standard experimental protocols. The length of ncRNAs is also taken into account in computational stud-

ies, and it will be used throughout our work to distinguish RNA sequences and structures in the different datasets considered.

The function of lncRNAs includes a role in higher-order chromosomal dynamics, telomere biology, and subcellular structural organization [11, 27]. Some lncRNAs play key regulatory and functional roles in the gene expression program of the cell. One of the vital functions is to act as ribozymes. Examples of naturally occurring ribozymes include group I and group II introns—RNase P and the hammerhead. The group, I and group II introns are usually 200 – 600nt long, catalyzing RNA splicing [65]. Many sncRNAs also contribute to the realization of similar biological functions. For example, small interfering RNAs contribute to gene regulation, transposon control and vital defence. microRNAs (miRNAs) participate in the post-transcriptional gene regulation, miRNAs, PIWI-interacting RNAs (piRNAs)) and promoter-associated RNAs (PARs) contribute to the gene regulation. More recently, many discoveries revealed several ncRNAs implicated in cancer growth and MCL-1 expression regulation [159, 205]. Those examples include ncRNAs from different classes, miRNAs, small nucleolar RNAs (snoRNAs) and T-UCR, all associated with a specific disease [49, 159].

There are also other classes of ncRNAs such as aptamers and riboswitches that have also been observed in nature. Aptamers are ncRNAs that can bind to other specified targets, whose nature is highly diverse. They range from small molecules to larger molecules. In some contexts, aptamers are termed riboswitches; for example, when their function is to sense the presence of an associated metabolite to cause a specific cis-reaction and/or cis-regulation of subordinated functional pathways [213].

In sum, lnc/snc-RNAs contribute to the realization of various biological functions, and they are mostly distinguishable based on their length and functions. But, their functions allow us to distinguish them better. In the next section, we provide some of the recent advancements in the techniques used to identify functional ncRNAs.

1.3 RECENT ADVANCEMENTS IN DETERMINING NCRNA FUNCTIONS

Most of the previously mentioned functions of ncRNAs are identified using gene targeting techniques, a well-known set of techniques used to investigate protein functions [163]. In addition, experimental approaches are used to define ncRNA functions. With the recent advancements in genome engineering, a method

such as clustered regularly interspaced short palindromic repeats (CRISPR) has been employed to tag lncRNAs, allowing to capture specific RNA-protein complexes assembled *in vivo*. This section aims at providing an overview of different techniques used to determine ncRNA functions.

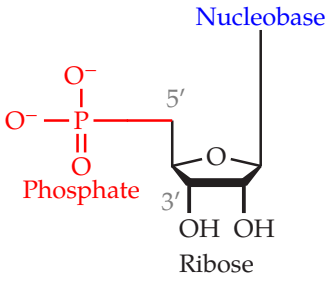
The CRISPR [8] was described by Barrangou and his collaborators in 2007 as a distinctive genome feature of most bacteria and archaea and thought to be involved in resistance to bacteriophages. It is an adaptive defence system against viruses and plasmid intrusions. When a successful defence takes place, the system updates information about the intruder's genetic material. This update will then allow the system's host to identify its enemy, making it robust and durable in the future. The information about the intruder's genetic material is stored in short repeating stretches of RNA, which can, in the case of a new intrusion, be incorporated into a carrier protein(CAS). The capacities of the CRISPR/CAS9 of selectively destroying foreign DNA/RNA and editing the genome was identified by Li et al. [108], and it was turned into methods allowing to alter and edit single genes within genomes selectively. The same technology is also successfully applied to animal cell lines [85, 91, 204] and industrial plants [109, 186].

Another method of systematic evolution of ligands by exponential enrichment (SELEX) [196] introduced by Tuerk in the early 1990s offers the possibility of enriching stretches of RNA that can bind a certain target. The method relies on mechanisms usually ascribed to the process of evolution, that is, variation, selection, and replication. A pool of RNAs that are entirely randomized at specific positions is subjected to selection for binding, in this case to GP43 on nitrocellulose filters. The selected RNAs are amplified as double-stranded DNA competent for subsequent *in vitro* transcription. This newly transcribed RNA is enriched for better binding sequences and is then subjected to selection to begin the next cycle. Multiple rounds of enrichment result in the exponential increase of the best binding ligands until they dominate the population of sequences. SELEX has given rise to numerous synthetic aptamers with different targets in its application. They have been subject to a further extension towards inclusion into regulative RNA entities.

More recently, increased types of ncRNAs have been detected and identified by the development of next-generation sequencing [206], which can be roughly divided into the process sections

of sample preprocessing, library preparation, sequencing, and bioinformatics.

The functions of many ncRNAs are dependent on their high-level structures, which often depend on lower-levels sequence and secondary structures. Knowing the structure of an ncRNA plays a vital role in probing its function. For example, Peter Flor and his collaborators used structure information to interpret experiments related to the mechanism of RNA function [56]. Or, Yoon et al. suggested new experiments based on RNA secondary structure in yeast to probe RNA functions [96]. Therefore, understanding even the secondary structure alone can assist both of these examples. In the following section, we provide a biochemical definition of the elementary building blocks of ncRNAs, which are the nucleotides A, U, G and C. In addition, we will provide an overview of the different nucleotide interactions involved during the formation of their secondary structures.

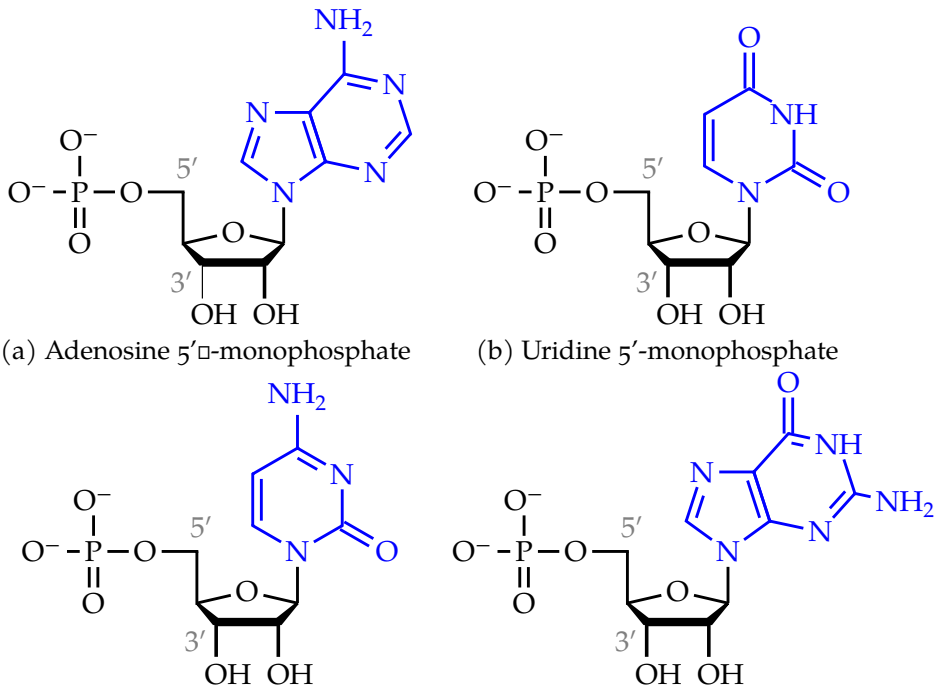


Structure of an RNA nucleotide

1.4 BIOCHEMISTRY OF RNA MOLECULES

So far, we have provided a biological motivation for studying non-coding RNA as an independent entity. The discovery of new ncRNAs functions has emerged through intensive experimental studies and with recent advanced techniques in next-generation sequencing. Several examples demonstrated the importance of the ncRNA structures in the probing process of new functions. The process in which RNA sequences are mapped to their corresponding structures is called RNA folding. In nature, this process is thought to be hierarchical [16, 194]. Nucleotides form a chain given their sequence of bases (primary structure); RNAs fold into secondary structures, such as stem-loops and helices, before folding into higher-level (tertiary and quaternary) structures. Our work is restricted here to the secondary level of an RNA structure, i.e., the set of canonical pairs. This section provides a biochemical definition of different nucleotides and base-pair interactions involved in the secondary structure folding of RNA molecules.

Chemically, each nucleotide in RNA molecules consists of a phosphate residue, a pentose sugar and a nucleobase. The typical chemical structure of a nucleotide is depicted on the right side of the page. Figure 1.1 illustrates the chemical structure of each of the four different nucleobases found in RNA (A, C, G and U). A nucleotide is a nucleoside which has a (mono, di, trip) phosphate residue bound to its 5'-carbon atom. By convention, the carbon



(a) Adenosine 5′□-monophosphate (b) Uridine 5′-monophosphate

(c) Cytidine 5′□-monophosphate (d) Guanosine 5′□-monophosphate

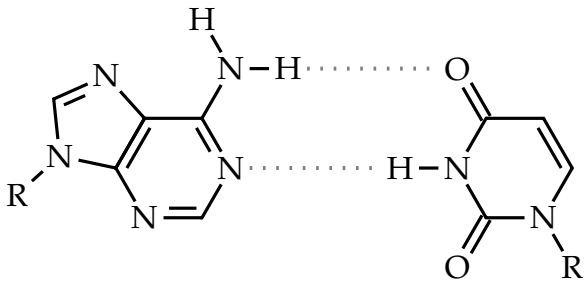
Figure 1.1: **RNA nucleotides.** Adenine and guanine belongs into the chemical class of purine molecules and the uracil and thymine in the class of pyrimidines

atoms of the pentose sugar in nucleotides are numbered with *primes*.

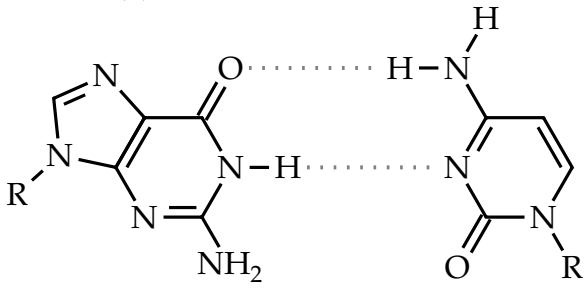
At the lowest level, RNA molecules are simply represented as a list of nucleobase characters. The 5′-3′ phosphodiester bonds attach the different nucleotides composing the RNA molecule between ribose to form the primary structure of RNA. The chain direction is conventionally designed as 5′ to 3′ (i.e. from 5′-phosphate first sugar backbone to the 3′-hydroxyl last sugar in the sequence).

In contrast to the RNA primary structure, the secondary structure consists of a list of nucleobase pairs, and the hydrogen bonds between the bases form base-pairs. Different interactions are possible between the bases depending on the structure level considered. At the secondary level, we have the Watson-Crick (or canonical) pairs [154, 166] (A-U and G-C), the Wobble (or non-canonical) (G-U) pairs that occur with reduced frequency. Figure 1.2 shows the chemical base-pairs for the Watson-Crick and Wobble interactions.

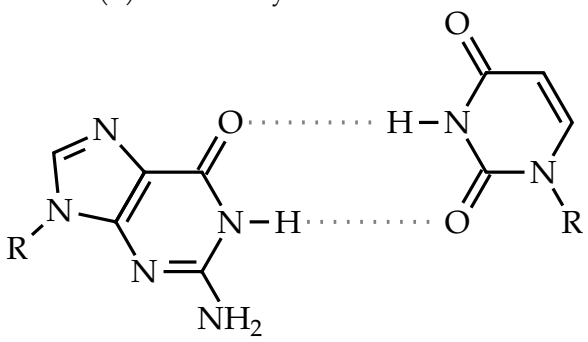
In addition to the Watson-Crick (WC) and wobble interactions, we also find crossing or pseudoknotted interactions in natural RNA that play vital roles in realizing biological functions, e.g. ribo-



(a) Adenine-Uracil Interaction



(b) Guanine-Cytosine Interaction

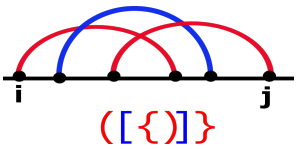


(c) Guanine-Uracil Interaction

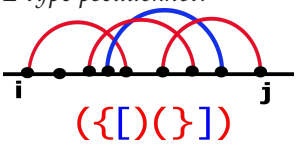
Figure 1.2: **RNA base-pair interactions**. (a) and (b) are commonly know as Watson-Crick base-pairs. (c) is the wobble base-pair. Hydrogen bonds are indicated as grey dashed lines. Substitution of bold hydrogen residues with ribose 5-phosphate yields the corresponding nucleotides found in RNA molecules.

somal frame-shifting [63], regulation of translation and splicing [42], or the binding of small molecules [64, 98, 182]. Although the pseudoknots are not considered in the computational folding tool we propose in Chapter 3, they are essential to evaluate the performance of the computational tool we will introduce in Chapter 5 for RNA design. This section also presents different pseudoknot patterns found in natural RNA and emphasizes the one considered in our work.

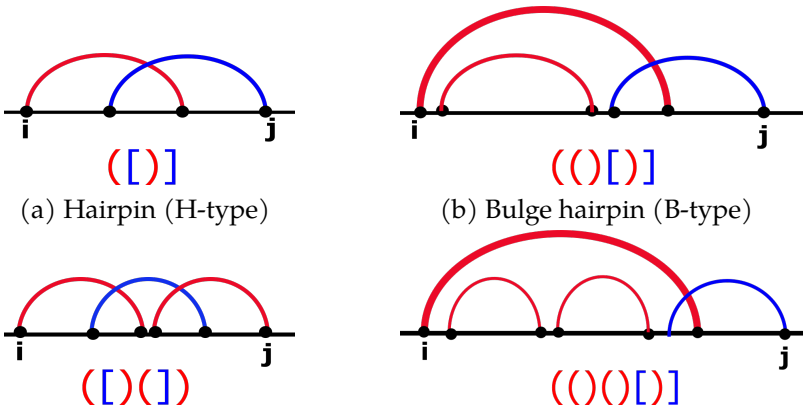
Pseudoknots occur when two WC, wobble or non-canonical interactions cross each other [210]. Even though pseudoknots are often considered the beginning of the interaction between the secondary and tertiary levels of RNA structures, they account in this work as part of the secondary structure. The restriction to only crossing WC and wobble base pairs contrasts the other tertiary interactions, which may include a broader class of interactions. Many pseudoknot patterns have been identified in natural RNAs. Most occurring pseudoknot patterns tend to be relatively simple in the sense that their crossings are not interlaced and may be viewed as superpositions of two nested secondary structures (bi-secondary structures) [76]. The simplest pattern is often termed as Hairpin-type or H-type (see Figure 1.3a). More complex forms of H-type pseudoknot are bulge hairpin (B-type) or complex hairpin (cH-type). H-type and K-type pseudoknots are the most frequent pseudoknots, but more complex but less frequent pseudoknots are possible such as M-type and L-type pseudoknots (See the figure on the right side of the page) [107]. The four types of pseudoknot patterns considered in Chapter 5 are depicted in Figure 1.3.



L-type pseudoknot.



M-type pseudoknot



(a) Hairpin (H-type)
(b) Bulge hairpin (B-type)
(c) Kissing hairpin (K-type)
(d) Complex hairpin (cH-type)

Figure 1.3: **Pseudoknot patterns found in the `PseudoBase++`**. For each pseudoknot patterns, the different rows represent respectively the circular and the dotbracket shape representations. The B-type and cH-type are more complex forms of H-type. The full complexity order is H-type< B-type<cH-type<K-type.

Considering pseudoknots in designing functional RNAs is vital given their role in realizing biological functions. Nevertheless, computationally folding an RNA molecule with arbitrary pseudoknot patterns is non-deterministic polynomial-time (NP)-complete [118]. Solving this problem is a prerequisite for RNA design and is still a real challenge, not only because of the computational constraint but also the experimental energy measurements of the pseudoknot interactions. In most cases, existing computational tools are restricted to a specific pseudoknot pattern and are based on approximated energy parameters [67].

In the context of this work, we consider two main secondary structure definitions: a pseudoknot-free one in which only canonical interactions with no crossing pairs are allowed and a second one where canonical interactions with possible crossing pairs are permitted. The following section will provide formal definitions and the framework in which we can computationally study the folding of the secondary structure of ncRNAs.

1.5 BIOINFORMATIC DEFINITIONS

We provided in the previous sections the biological motivations and biochemical concepts that support the computation methods studied in the thesis. In order to computationally study and analyse RNA molecules, a more formal representation of RNAs and bioinformatic definitions are required. We provide in this sec-

tion formal definitions and concepts that will support the result presented in this thesis.

1.5.1 *Structural definitions*

This thesis focuses on computational folding and inverse folding methods of the secondary structure of RNA molecules. The secondary structure, in most cases, is computed for a given RNA sequence. Along the thesis, ϕ will represent an RNA sequence of a fixed length L and $\mathcal{S}$ its corresponding structure. This subsection provides formal definitions of ϕ, $\mathcal{S}$ and the structural properties of $\mathcal{S}$. We will assume the same definitions in the different tools reviewed in Chapter 2, Chapter 4, which also supports the results presented in Chapter 3 and Chapter 5.

Definition 1 (RNA sequence)**.** More formally, ϕ consists of an ordered sequence of nucleotides that can be represented as:

$$\phi = (\phi_1, ..., \phi_L), \tag{1.1}$$

where $\phi_i \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$ for $i \in \{1 \dots L\}$. ϕ is often known as the primary structure of RNA.

Definition 2 (RNA pseudoknot-free secondary structure)**.** Given an RNA sequence $\phi \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}^L$, let $\mathcal{D} = \{(i, j) : i < j\}$ be the list of possible pairing positions over the sequence ϕ. A pseudoknot-free secondary structure $\mathcal{S} \subset \mathcal{D}$ of such sequence ϕ is a list of base-pairs with the following constraints [79, 81]:

1. A nucleotide (sequence position) can only belong to a single pair, i.e. $\forall (i, j), (k, l) \in \mathcal{S}$ with $i < k$: $i = k \Rightarrow j = l$.
2. Paired bases must be separated by at least three unpaired nucleotides. i.e. $\forall (i, j) \in \mathcal{S} \Rightarrow j - i \geq 3$.
3. There are no pseudoknots, i.e. $\nexists (i, j), (k, l) \in \mathcal{S}$ with $i < k < j < l$,
4. The base-pairs consist exclusively of Watson–Crick (C–G and A–U) pairs and Wobble (G–U) pairs. i.e. $\forall (i, j) \in \mathcal{S} \Rightarrow \phi_i \phi_j \in \{\mathrm{GC}, \mathrm{CG}, \mathrm{AU}, \mathrm{UA}, \mathrm{GU}, \mathrm{UG}\}$,

Therefore, RNA secondary structures can be thought as planar graphs than can be more or less easily drawn on a plan.

Definition 3 (Secondary structure representation)**.** A graphical way of representing an RNA secondary structure. There are several representations of $\mathcal{S}$.

- Dot-bracket (or string) representation: In this representation, the secondary structure $\mathcal{S}$ is compactly stored in a string σ consisting of dots and matching brackets. i.e. σ is a string of length L over the alphabet $\Delta_\sigma = \{(,), [,], \{,\}, <, >, .\}$ where, at each unpaired positions we have a dot '.' at the corresponding string position, and $\forall (i,j) \in \mathcal{S}$, we have an opening bracket at position σ_i and a closing bracket at position σ_j. We denote σ the string representation of the structure $\mathcal{S}$. Figure 1.4D shows an example of a string representation.

- Planar representation: it is the common way of representing an RNA secondary structure in which $\mathcal{S}$ is presented as a graph with each vertex representing a nucleotide and an edge connecting consecutive nucleotides and base-pairs (See Figure 1.4B).

- Circular (or circle) representation: similar to planar representation, $\mathcal{S}$ is a graph but drawn in the plane in such a way that all vertices are arranged on a circle, and the edges representing base-pairs lie inside the circle. In a pseudoknot-free secondary structure circular representation, the edges do not intersect (See Figure 1.4A).

- Linear representation: In this representation, $\mathcal{S}$ is a graph in which the nucleotides are arranged consecutively in a line and the edges representing base-pairs form semi-circle that do not intersect for pseudoknot-free structure (See Figure 1.4C).

- Mountain representation: it is mainly used for representing large structures. $\mathcal{S}$ is presented in a two-dimensional graph, in which the x-coordinate is the position i of the nucleotide in the sequence ϕ and the y-coordinate the number $m(i)$ of base-pairs that enclose nucleotide i.

- Tree representation: $\mathcal{S}$ is drawn as a tree in which internal nodes are the base pairing positions, and the leaves are the unpaired positions. The dot-bracket representation is also often considered as a tree represented by a string of parenthesis (base-pairs) and dots for the leaf nodes (unpaired nucleotides).

- Shapiro representation: it allows representing the different elements composing $\mathcal{S}$ by single matching brackets, and

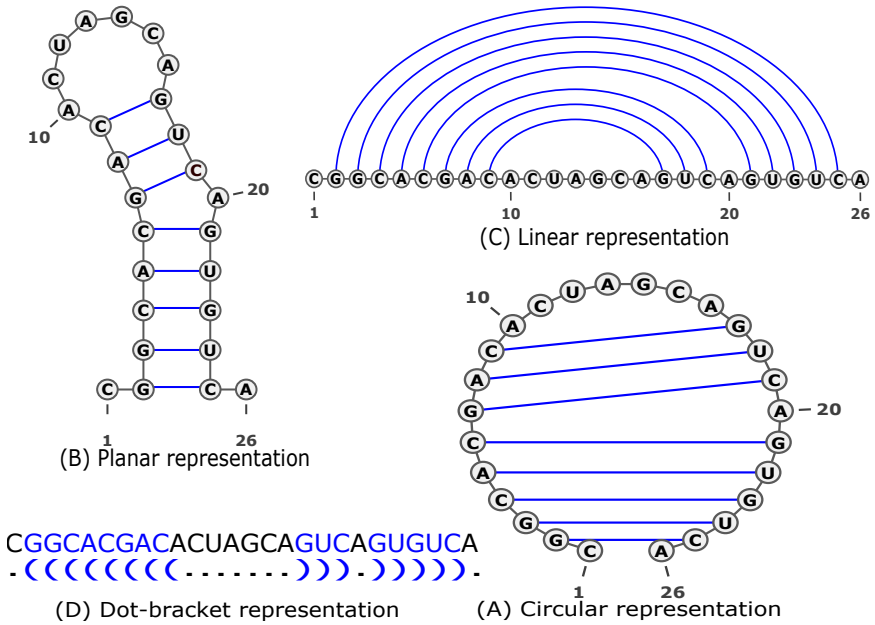
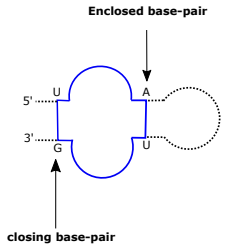


Figure 1.4: **Different secondary structure representations of a random generated RNA sequence.** The MFE structure is predicted using `RNAfold` from the `ViennaRNA Package` [112]. The representation were then drawn using `VARNA` [31]

the components are labelled with H(Hairpin), B(Bulge), I (interior loop), M (multi-loop) and S (stacking loop) [169].

Figure 1.4 shows some examples of RNA secondary structure representation. For graphical illustrating examples in the thesis, we will mostly use the planar representation, and for computational methods, we will use the dot-bracket representation for simplicity.



An example of closing and enclosed base-pairs of an interior loop.

Definition 4 (Secondary structure loop)**.** There exists a unique decomposition of $\mathcal{S}$ into a set of n loops $\mathbb{L}_{\phi,\mathcal{S}}$, where loops are the faces of its planar drawing. Each loop $\mathcal{L} \in \mathbb{L}_{\phi,\mathcal{S}}$ is characterised by its length l (the number of unpaired nucleotides in the loop) and its degree d (the number of base-pairs delimiting the loop, including the closing loop pair).

By definition, $\forall \mathcal{L} \in \mathbb{L}_{\phi,\mathcal{S}} \Rightarrow \mathcal{L} = \mathcal{L}_p \cup \mathcal{L}_u$ where $\mathcal{L}_p$ and $\mathcal{L}_u$ denote respectively the set of loop base-pairs and the unpaired positions. $\mathcal{L}_p$ contains only one closing loop and the rest are enclosed base-pairs. We say $(i,j) \in \mathcal{L}_p$ is a closing pair if and only if $\forall \mathcal{L}_p \ni (i',j') \neq (i,j) : i < i' < j' < j$.

1. Interior loop: a loop with degree $d = 2$ i.e $|\mathcal{L}_p| = 2$ and $\mathcal{L}_u \subset \{1, 2, \dots L\} \cup \emptyset$.

2. Stacking pair: an interior loop of length $l = 0$ i.e. $|\mathcal{L}_p| = 2$ and $\mathcal{L}_u = \emptyset$.

3. Hairpin Loop: Any loop of degree $d = 1$ and length $l \geq 3$. i.e $|\mathcal{L}_p| = 1$ and $\mathcal{L}_u \neq \emptyset$.

4. Bulge loop: a special case of interior loop in which there are unpaired bases only on one side. i.e $\mathcal{L}_p = \{(i_1, j_1), (i_2, j_2)\}$ with $i_1 \neq i_2, j_1 \neq j_2$ one of the following assumption holds:
   - If $\exists i' \in \mathcal{L}_u : i_1 < i' < j_2 \Rightarrow \nexists k' \in \mathcal{L}_u : i_2 < k' < j_2$
   - If $\exists k' \in \mathcal{L}_u : i_2 < k' < j_2 \Rightarrow \nexists i' \in \mathcal{L}_u : i_1 < i' < j_1$

5. Multi-loop: Any loop with degree $d > 2$ i.e. $|\mathcal{L}i_p| \geq 3$ and $\mathcal{L}_u \neq \emptyset$.

6. Exterior loop: a loop in which all the positions are not interior of any pair i.e. $\mathcal{L}_p = \emptyset$ and $\mathcal{L}_u \neq \emptyset$.

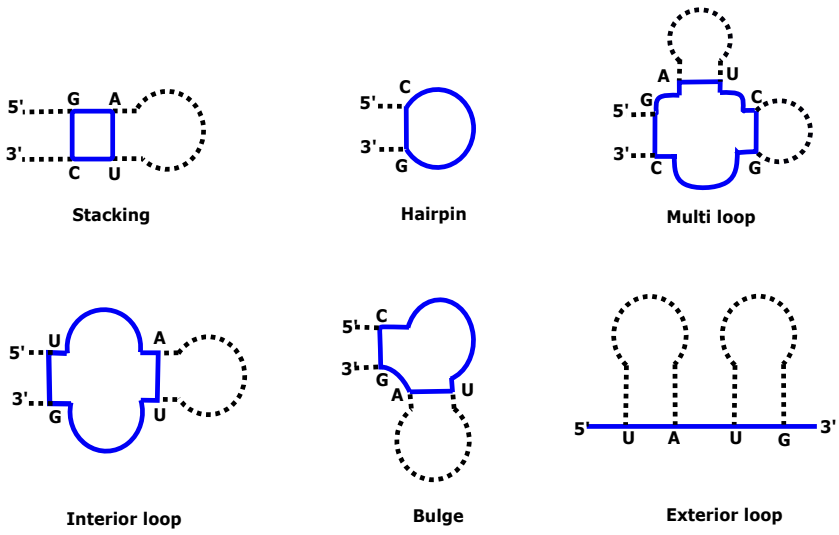


Figure 1.5: RNA **secondary structure loop decomposition**. Each loop is highlighted in blue.

Definition 5 (Free energy of an RNA secondary structure)**.** Given the loop set $\mathbb{L}_{\phi,\mathcal{S}}$, the free energy ΔG of $\mathcal{S}$ defines its thermodynamic stability. ΔG is the free energy difference with respect to the completely unfolded state [195]. $\Delta G(\mathcal{S}, \phi)$ is computed using

the additivity principle [34], by summing up the energies of its constituent loops.

$$\Delta G(\mathcal{S}, \phi) = \sum_{\mathcal{L} \in \mathbb{L}_{\mathcal{S},\phi}} \Delta G(\mathcal{L}) \tag{1.2}$$

Many models allow for computing the free energies of those constituent loops, but the dominant is the nearest-neighbor loop energy model [198]. This model associates tabulated free energy values to loop types and nucleotide compositions; the Turner2004 [123] is one of the most widely used parameter sets.

The free energy of each given loop $\mathcal{L}$ is expressed as

$$\Delta G(\mathcal{L}) = \Delta H - T\Delta S \leq 0 \tag{1.3}$$

where ΔH is the (pressure- and volume-dependent) enthalpy change, T the absolute temperature and ΔS the entropy change. The dominant stabilizing effect is attributed to consecutive base-pairs (The stacking loops), whereas long unpaired regions enclosed between base-pairs have destabilizing effects [58, 81]. As a simplified example, the destabilizing free energy contribution $\Delta G(\mathcal{L}_m)$ of a multiloop $\mathcal{L}_m$ as seen in Figure 1.5C is modelled as:

$$\Delta G(\mathcal{L}_m) = \Delta G_{\text{init}} + b\Delta G_{\text{branch}} + u\Delta G_{\text{unpaired}} \tag{1.4}$$

where b is the number of all surrounding base-pairs and u the number of base-pairs [36].

In addition to the definitions mentioned above, we have various properties of an RNA sequence such as structural diversity, positional entropy, structures with maximal expected accuracy, or the density of states. An extensive summary of all possible properties and the history of algorithms is reviewed by Lorenz [115].

The structure decomposition and the tabulated energy parameter sets allow an efficient dynamic programming algorithm to determine a sequence's secondary structure in the entire structure space. Several programs implementing algorithms will enable the computation of these properties efficiently. The thesis gives a literature review of such tools in Chapter 2.

1.5.2 *Thermodynamic definitions*

A common way to computationally address the RNA folding problem is to consider a dynamic system of structures (the states of

the system). Given enough time, a sequence ϕ will form every possible structure Σ_ϕ. For each structure $\mathcal{S} \in \Sigma_\phi$, there is a probability of observing it at a given time. This subsection defines RNA folding thermodynamic properties such as structural ensemble, partition function, Boltzmann probability of a structure $\mathcal{S}$, and the others that derive from them, the base-pair probability and the most probable secondary structure.

The folding tools such as `RNAfold`, `LinearFold` used in this thesis use the same thermodynamic definitions. However, some computational folding methods do not rely on a thermodynamic model. For example, Chapter 2 presents a literature review of such tools.

Definition 6 (Structure Ensemble)**.** For a given RNA sequence ϕ, the set of all pseudoknot-free secondary structures with their corresponding energies is called the structure ensemble Σ_ϕ of ϕ or Boltzmann ensemble. We write:

$$\Sigma_\phi = \{\mathcal{S} | \mathcal{S} \text{ is a secondary structure of } \phi\}$$

According to the nearest neighbor energy model, all possible secondary structures of a given RNA sequence do not have the same energy. Since each structure has a unique decomposition, each structure has its own energy but different structures can have the same energy.

Definition 7 (Partition function of RNA)**.** Given the free energy change $\Delta G(\mathcal{S})$ of a structure $\mathcal{S}$, the partition function $Z(\Sigma_\phi)$ is defined on the Boltzmann ensemble (or structure ensemble) of all possible structures of a given sequence ϕ and we write:

$$Z(\Sigma_\phi) = \sum_{\mathcal{S} \in \Sigma_\phi} \exp(-\beta \Delta G(\mathcal{S}, \phi)) \tag{1.5}$$

Where, $\beta = (RT)^{-1}$ with R the ideal gas constant, and T the temperature.

Definition 8 (Secondary structure probability)**.** How probable is an RNA secondary structure $\mathcal{S} \in \Sigma_\phi$ for the sequence ϕ? Given the free energy change $\Delta G(\mathcal{S})$ of a structure $\mathcal{S}$, the boltzmann distribution describes the structure's probability at constant temperature T among all other possible structure of the same sequence ϕ. The probability $p(\mathcal{S} | \phi)$ depends on the free energy $\Delta G(\mathcal{S})$, the lower the more probable. We write:

$$p(\mathcal{S} | \phi) = \frac{\exp(-\beta \Delta G(\mathcal{S}, \phi))}{Z} \tag{1.6}$$

where, Z is the partition function and $\beta = (RT)^{-1}$ the thermal constant.

Definition 9 (MFE secondary structure)**.** To predict biologically relevant structures, most computational methods search for structures that minimize the free energy. For a given sequence ϕ, let Σ_ϕ be the secondary structure ensemble of ϕ. The minimum free energy structure $\mathcal{S}_{MFE}$ is the structure with the lowest probability $p(\mathcal{S}|\phi)$ i.e. the most stable conformation in the thermodynamic equilibrium. We write:

$$\mathcal{S}^{MFE}(\phi) = \arg\min_{\mathcal{S}\in\Sigma_\phi} \Delta G(\mathcal{S}, \phi) \tag{1.7}$$

Definition 10 (Base-pair probability)**.** Let $\phi = (\phi_i)_{1\leq i\leq L}$ be an RNA sequence. The base-pair probability matrix $\mathbf{P}(\phi)$ quantifies the equilibrium structural features of the ensemble Σ_ϕ, with entries $P_{i,j}(\phi) \in [0, 1]$ defines as follows:

$$P_{i,j}(\phi) = \sum_{\mathcal{S}\in\Sigma_\phi} p(\mathcal{S}|\phi) S_{i,j}(\mathcal{S}) \tag{1.8}$$

$P_{i,j}(\phi)$ corresponds to the probability that base-pair $i.j$ forms at the equilibrium. $\mathbf{S}(\mathcal{S})$ is the structure matrix with entries $S_{i,j} \in \{0, 1\}$. If the structure $\mathcal{S}$ contains pair (i, j), then $S_{i,j}(\mathcal{S}) = 1$ otherwise $S_{i,j}(\mathcal{S}) = 0$.

The base pair probabilities enable then a new view at the structure ensemble. Figure 1.6 shows an example of MFE structure and the base pair probability dot plot [1] of a tRNA. A square at row i and column j indicates a base pair. The area of a square in the upper right half of the matrix is proportional to the base pair probability (i, j) within the equilibrium ensemble. The lower left half shows all pairs belonging to the MFE structure. While the MFE consists of hairpins, bulge and stacking, several different loops are visualized in the pair probabilities, which leads to several local minima with different shapes.

1 computed using `RNAfold -p`

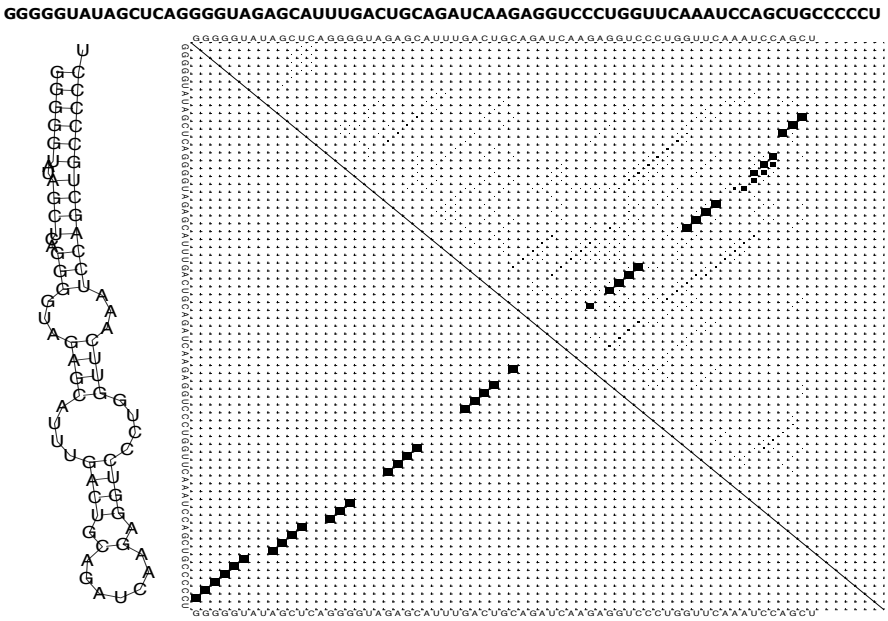


Figure 1.6: **Base-pair probability matrix** of a tRNA sequence computed using `RNAfold 2.4.13`. The MFE structure is depicted on the left and the sequence on top. The frequency of the MFE structure in the structural ensemble Σ_ϕ is 0.116. The dot plot on the right shows the pair probabilities within the equilibrium ensemble as (72×72)-matrix and is an excellent way to visualize structural alternatives.

The definitions mentioned above provide us with a necessary framework enabling us to compute the MFE secondary structure within the equilibrium ensemble Σ_ϕ. Several implementations of these definitions have been suggested [112, 148, 218], and they are available as an application programming interface (API). In the context of this work, we are not only interested in the MFE structure but, instead, we use some features of the existing computer libraries (e.g. the computation of the structure free energy) to predict an ensemble structure. The following section introduces some metrics used in this dissertation to compare RNA secondary structures and, eventually, the structure predictions produced by different tools.

1.5.3 *Structural distance definitions*

The validation of the results obtained in this thesis is purely empirical. We achieved this goal by comparing the predicted and expected structures for the folding tools. We use the PPV and the sensitivity's statistical properties for the benchmark results

presented in Chapter 3. For the inverse folding tools, we compare the MFE structure of the designed sequence to the target structure. For that end, a rigorous definition of a measure of similarities between two structures is needed. This subsection defines the different similarity measurements used throughout this work. In addition, it defines the objective functions used in our inverse folding presented in Chapter 5.

Definition 11 (The PPV)**.** It measures the fraction of correct base-pairs in the predicted structure and it is defined as follows:

$$PPV = \frac{TP}{TP + FP} \tag{1.9}$$

where TP and FP stand respectively for the number of correctly predicted base-pairs (true positives), and the number of wrongly predicted base-pairs (false positives).

Definition 12 (Sensitivity)**.** It measures the fraction of base-pairs in the accepted structure that are predicted.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{1.10}$$

where FN stands for the number of base-pairs not detected (false negatives).

Definition 13 (Base-pair distance)**.** Let σ_1 and σ_2 be two secondary structures in their string representation. The base-pair distance between σ_1 and σ_2 is defined as follows:

$$d_{bp}(\sigma_1, \sigma_2) = \sum_{i,j} A_{i,j}[\sigma_1] + A_{i,j}[\sigma_2] + 2 \times A_{i,j}[\sigma_1] A_{i,j}[\sigma_2], \tag{1.11}$$

where,

$$A_{i,j}[\sigma] = \begin{cases} 1 & \text{if } (i,j) \text{ is a base-pair in } \sigma \\ 0 & \text{otherwise} \end{cases}$$

Definition 14 (Hamming distance)**.** Let σ_1 and σ_2 be two secondary structures in their string representation. We define the hamming distance between σ_1 and σ_2, $d_h(\sigma_1, \sigma_2)$, to be the number of position where σ_1 and σ_2 differ.

$$d_h(\sigma_1, \sigma_2) = \sum_{i=1}^{L} S(\sigma_1^i, \sigma_2^i) \tag{1.12}$$

where,

$$S(\sigma_1^i, \sigma_2^j) = \begin{cases} 1 & \text{if } \sigma_1^i \neq \sigma_2^j) \\ 0 & \text{otherwise} \end{cases}$$

Definition 15 (Ensemble defect (ED) [219])**.** Given an RNA sequence ϕ of length L, the ensemble defect $\mathcal{D}_E$ is the expected base-pair distance between a target structure $\mathcal{S}^*$ and a random structure generated with respect to the Boltzmann probability distribution. It is defined as follows:

$$\begin{aligned} \mathcal{D}_E(\phi, \mathcal{S}^*) &= \sum_{\mathcal{S} \in \Sigma_\phi} p(\mathcal{S}|\phi) d_{bp}(\mathcal{S}, \mathcal{S}^*) \\ &= L - \sum_{1<i,j<L} P_{i,j}(\phi) S_{i,j}(\mathcal{S}^*) \end{aligned} \tag{1.13}$$

where $P_{i,j}$ is the base-pair probability matrix entrances, $d_{bp}((\mathcal{S}, \mathcal{S}^*))$ is the base pair distance between two structures, and $\mathbf{S}(\mathcal{S})$ is the structure matrix with entries $S_{i,j} \in \{0, 1\}$. If the structure $\mathcal{S}$ contains pair (i, j), then $S_{i,j}(\mathcal{S}) = 1$ otherwise $S_{i,j}(\mathcal{S}) = 0$.

Definition 16 (Normalized Energy Distance (NED))**.** the difference between the energy of a given sequence ϕ evaluated to fold into a target structure $\mathcal{S}^*$ and the minimum free energy of the sequence in its structural ensemble Σ_ϕ. The value is normalized over all the sequences in a given population P.

$$\mathcal{N}_E(\phi, \mathcal{S}^*) = [1 - \Delta\hat{E}(\mathcal{S}^*, \phi)]^q \ \forall q > 1 \tag{1.14}$$

where,

$$\Delta\hat{E}(\mathcal{S}^*, \phi) = \frac{\Delta E(\mathcal{S}^*, \phi)}{\sum_{s \in P} \Delta E(\mathcal{S}^*, s)} \tag{1.15}$$

and,

$$\Delta E(\mathcal{S}^*, \phi) = \Delta G(\mathcal{S}^*, \phi) - \arg\min_{\mathcal{S} \in \Sigma_\phi} \Delta G(\mathcal{S}, \phi) \tag{1.16}$$

Among the definitions mentioned above, 11 and 12 are used in Chapter 3 for the benchmark comparison. Whereas, definitions 13, 14, 15, 16 are used in Chapter 5 for both objective function and benchmark purposes. The following section provides a formal definition of the fitness landscape and some of its properties. It will mostly use 14 for both structure and sequence comparison.

1.5.4 *RNA folding map properties*

This work considers RNA molecule folding and inverse folding optimisation problems. In both cases, It is fundamental to define the fitness landscape notion. This subsection provides the formal definitions of the fitness landscape and examples related to the folding and inverse problem. Some properties such as neutrality, mutation mode or move operator are also provided. The size of the RNA structural ensemble has been analytically computed through tools developed by Stein and Waterman [185], and it yields an upper bound of $S_L \approx 1.48 \times L^{-\frac{3}{2}} 1.85^L$ structure vis-a-vis 4^L sequences. Compared to the total number of sequences, the number of structures is much smaller, which means there is a high possibility that many sequences fold into the same MFE secondary structure. In case that happens, we call the set of those sequences a neutral set. The fraction of such sequences defines the neutrality of a fitness landscape.

Definition 17 (Fitness landscape)**.** A fitness landscape $\mathfrak{L}$ results from the combination of three elements: a set of configurations $\mathcal{U}$, a cost or fitness function f, and a *move* operator ψ that induces a topology on the set of configurations. We write:

$$\mathfrak{L} = (\mathcal{G}_f, f, \psi) \tag{1.17}$$

where $\mathcal{G}_f$ is the the landscape underlying the hypergraph whose vertices are the elements from $\mathcal{U}$ labelled with values given by f, and whose edges are specified by the move operator ψ.

The fitness function f assigns to each configuration $v \in \mathcal{U}$ a real value taken from an interval $\mathbb{I} \subset \mathbb{R}$ as follows:

$$f : \mathcal{U} \to \mathbb{I}$$

An example of fitness function in the case of inverse folding is defined in Chapter 5 (Section 5.1.2), which uses the hamming distance d_h and $\mathcal{U} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}^L$. But in this case, the fitness defined in the structural space Σ_ϕ. i.e. we have an intermediate folding function $\Delta G(\mathcal{S}, \phi)$, mapping any sequence $\phi \in \mathcal{U}$ to an MFE secondary structure.

The move (or mutation) operator ψ defines the relationship between the configuration from $\mathcal{U}$ in the following way:

$$\psi \colon \mathcal{U} \to \mathcal{U}$$

Definition 18 (Mutation mode)**.** Let $\phi, \phi' \in \mathcal{U} = \{A, C, G, U\}^L$, be two RNA sequences. ϕ' is said to be an n-point mutation of ϕ if it differs from ϕ at n nucleotides; i.e. $d_h(\phi, \phi') = n$ where $d_h(.,.)$ is the hamming distance on $\{A, C, G, U\}^L$.

A mutation mode is a random variable U taking values in $\{1, \dots, L\}$. $P(U = n)$ is defined as the probability that, exactly n nucleotides, selected uniformly at random undergo point mutation during a mutation event. U can generally be any probability distribution.

Definition 19 (Neutral set of RNA sequences)**.** For a give fitness landscape $\mathfrak{L} = (\mathcal{G}_f, f, \psi)$, with $\mathcal{U} = \{A, C, G, U\}^L$, two RNA sequence ϕ_1 and ϕ_2 are set to be neutral $\iff f(\phi_1) = f(\phi_2)$. We call a set $\Gamma \subset \mathcal{U}$ of all such RNA sequences a neutral set. In the case of inverse folding, ϕ_1 and ϕ_2 are neutral if they share the same MFE secondary structure. In contrast, ϕ_1 and ϕ_2 have the same free energy in the folding problem context.

Definition 20 (Neutral Network)**.** Let $\mathcal{G}(\mathcal{U}, E)$ be a connected graph in which vertices are all in the neutral sequence set Γ (i.e. $\mathcal{U} \subset \Gamma$). $\mathcal{G}$ is said to be a neutral network $\iff \forall e(v_i, v_j) \in E, v_i, v_j$ differ by a single nucleotide (i.e. $d_h(v_i, v_j) = 1$).

We provided in this subsection a general definition of a fitness landscape with examples related to computational RNA folding and inverse folding. Now that we have all the ingredients to computationally study the folding and the inverse folding of RNA molecule, we are left with the definition of some computational techniques used in our proposed tools. Our contributions rely on two well-known techniques of algorithms: the FFT for the folding mechanism and the EA for the inverse folding. An overview of both techniques is provided in the following section.

1.5.5 *The fast Fourier transform (FFT) and evolutionary algorithm (EA) applied to RNA bioinformatics*

The computational results present in this work rely on two well-known techniques: the FFT and EA. Both approaches have already been studied and have found many applications, including the computational folding and inverse folding of ncRNA. This section gives a short overview of the two concepts.

A FFT is an algorithm that computes the distcrete Fourier transform (DFT) of a sequence or its inverse (Inverse discrete Fourier transform (IDFT)). Fourier analysis converts a signal from its

original domain (often time or space) to a representation in the frequency domain and vice versa. The DFT is obtained by decomposing a sequence of values into components of different frequencies.

More formally, let $\{x_k\} := x_0, \dots, x_{L-1}$ be a sequence of L complex numbers, the DFT transforms the sequence $\{x_k\}$ into another sequence of L complex numbers $\{X_k\} := X_0, \dots, X_{L-1}$ defined as follows:

$$X_k = \sum_{n=0}^{L-1} x_n e^{-i2\pi kn/N} \tag{1.18}$$

The direct evaluation of Equation 1.18 will require $O(L^2)$ operations because there are L outputs of X_k, and each of them requires a sum of L terms. A FFT is, therefore, any approach allowing to compute the same results in $O(L \log L)$ operations [92].

Let x and y be two sequences of length L and let X and Y be their respective DFTs. The correlation c_k between sequence x and y with the positional lag of k sites is defined as follows:

$$c_k = \sum_{1<n<L, 1<n+k<L} x_n y_{n+k} \tag{1.19}$$

It is known that the correlation c_k can be expressed in terms of the DFT. We write:

$$c_k \Leftrightarrow X_n^* \cdot Y_n \tag{1.20}$$

Where the asterisk denotes complex conjugation. That means we simply need to compute the DFT X_n and Y_n. Therefore, we can compute correlations c_k using the FFT as follows: FFT the two sequences, multiply one resulting transform by the complex conjugate of the other, and inverse transform the product.

Similar to Equation 1.18, the direct evaluation of c_k requires $O(L^2)$ operations and taking advantage of the FFT reduces it to $O(L \log L)$ operations. Several FFT algorithms have been implemented to speed up the computation of the DFT but so far, the most commonly used is the Cooley–Tukey algorithm [25].

The same idea has been applied in the context of RNA bioinformatics, where the two sequences of complex numbers can be thought of as two data sets of real numbers encoding the RNA sequences information. And the correlation c_k measures the homologous region in the two RNA sequences [94]. In contrast to

Katoh and his collaborators [94], we use the FFT to rapidly identify the largest stems of an RNA sequence. Thanks to the FFT which allows us to efficiently predict the fast-folding pathways of RNA molecules (See Chapter 3) within a reasonable CPU time.

The EA is another well-known heuristic approach, especially when dealing with problems in which less information about the fitness landscape is provided or when there is no exact algorithm in polynomial for such problems. The EA approach is inspired by evolutionary systems. In the 1950s and the 1960s, several computer scientists already independently studied evolutionary systems with the idea that evolution could be used as an optimization tool for engineering problems [131]. The picture in all these systems was to evolve a population of candidate solutions to a given situation, using operators inspired by natural genetic variation and natural selection.

Since the genetic algorithm (or more generally EA) was proposed by John Holland [82] in the early 1970s, it has emerged as a popular search heuristic. It has found application in many disciplines that deal with complex landscape optimization problems, e.g. RNA folding [132, 211] and inverse RNA folding [47, 48, 189].

EAs form a class of heuristic search methods based on a particular algorithmic framework whose main components are the variation operators (mutation and recombination or crossover) and the selection operators (parent selection and survivor selection). The general evolutionary algorithm framework is depicted in Figure 1.7. In most of the EA implementations, the solutions are encoded in the form of genomes (array of elements). The simplest form of EA typically involves two types of operators: selection and mutation (single point).

- **selection**: the operator consists of selecting solutions in the population for reproduction. The fitter the solution, the more times as likely it is selected to reproduce. This operator often requires a fitness function evaluation.

- **mutation**: the operator allows generating new solutions in the population. It randomly flips or permutes some element positions in a genome solution. For example, if we encode the solutions in a binary string, the solution 00000100 might be mutated in its second position to yield 01000100. Mutation can occur at each bit position in a string with some probability, usually very small (e.g. 0.001 for a sequence of length 50).

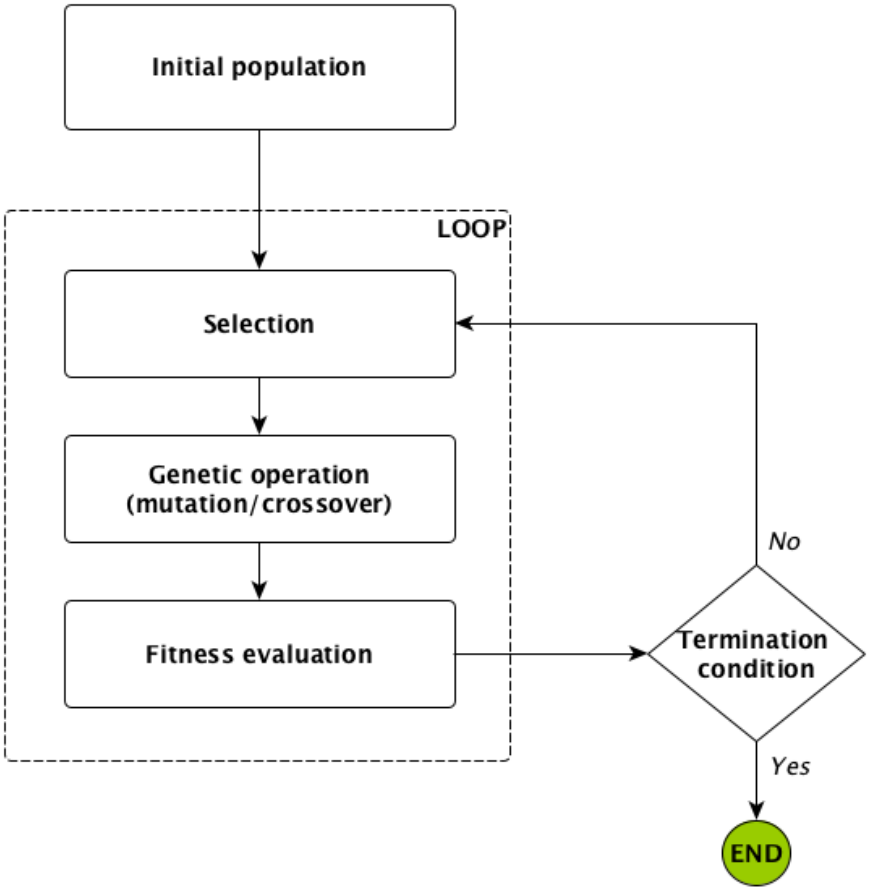


Figure 1.7: **Evolutionary algorithm flow diagram**. The algorithm initializes a population of candidate solutions and then loops over the three genetic operations until the termination criteria are satisfied.

In a more complex configuration, we can have a crossover operator that plays almost the same role as mutation, which generates new solutions in the population. In contrast to the mutation operator, the crossover randomly chooses a locus and exchanges the subsequence solutions before and after that locus between two solutions to create two offspring solutions.

In the context of this work, we use the simplest form of EA, in which we did not consider a crossover operator. We implement the simplest EA framework with a mutation operator adapted to the Inverse folding problem, which results in an alternative computational tool named `aRNAque` (see Chapter 5)).

This section provided an overview of the two main tools used in this thesis, which are EA and FFT. The EA is implemented in the computational inverse folding tool we propose in Chapter 5,

and the FFT in the RNA folding tool that will be introduced in Chapter 3.

1.6 CONCLUSION AND OUTLINE OF THE THESIS

This introductory chapter presents nucleic acids in general and, in particular, a description of ncRNA and its chemical, biological, and algorithmic definitions. Those concepts with biological motivations constitute the basis of the thesis.

We organize the next part of the thesis into fives. The two first s are grouped into a first result part which only concerns RNA folding. The second part discusses inverse folding, and similarly to the first part, it contains two s. The last discusses the presented results and concludes by providing some limitations and possible future research directions.

In Part i, Chapter 2 provides a brief literature review on the existing computational methods for RNA folding. The review focuses on thermodynamic and machine learning methods such as `RNAfold`, `LinearFold` and `Mxfold`. We review some of the limitations of existing tools in Chapter 2, such as the computational time, and in some cases, the predicted thermodynamic structure does not match the native one. Chapter 3 presents our proposed folding tool called `RAFFT`, which aims at overcoming those limitations. `RAFFT` implements a novel heuristic to predict RNA secondary structure formation pathways that has two components: (i) a folding algorithm and (ii) a kinetic ansatz. This heuristic is inspired by the kinetic partitioning mechanism, by which molecules follow alternative folding pathways to their native structure, some much faster than others. `RAFFT` starts by generating an ensemble of concurrent folding pathways ending in multiple metastable structures, which contrasts with traditional thermodynamic approaches that find single structures with minimal free energies. When analyzing 50 predicted folds per sequence, we found near-native predictions for RNAs of length ≤ 200 nucleotides, matching the performance of current deep-learning-based structure prediction methods [160, 220]. `RAFFT` also acts as a folding kinetic ansatz, which we tested on two RNAs: the CFSE and a classic bi-stable sequence. For the CFSE, an ensemble of 68 distinct structures computed by `RAFFT` allowed us to produce complete folding kinetic trajectories. In contrast, known methods require evaluating millions of sub-optimal structures to achieve this result. For the second application, only 46 distinct structures were required

to reproduce the kinetics, whereas known methods required a sample of 20,000 structures.

Similar to the first part of the result, Part ii contains two chapters. Chapter 4 will briefly introduce the RNA design problem. It distinguishes the positive from the negative RNA design problem and reviews the current state of the art computational tools, especially those implementing evolutionary techniques. The existing tools present challenges when benchmarked on recent datasets such as `Eterna100`. Another limitation is that most existing tools do not consider the pseudoknot patterns in their designing process. In Chapter 5, we propose an improved evolutionary algorithm inspired by the Lévy flights. Like a Lévy flight, our tool, `aRNAque`, implements a Lévy mutation scheme that allows simultaneous search at all scales over the mutational landscape. New mutations often produce nearby sequences (one-point mutations) but occasionally generate mutant sequences far away in genotype space (macro-mutations). In `aRNAque`, the number of point mutations distribution at every step is taken to follow a Zipf distribution. The Lévy mutation scheme increases the diversity of designed RNA sequences and reduces the average number of evaluations of the evolutionary algorithm compared to the local search. The overall performance showed improved empirical results compared to existing tools through intensive benchmarks on both pseudoknot (the `PseudoBase++` dataset) and pseudoknot-free (the `Eterna100` dataset) datasets.

Finally, Chapter 7 presents a general conclusion, a discussion on the results obtained and some promising perspectives. It emphasizes the understanding of the Lévy mutation in the context of RNA design and the application of our results to evolutionary dynamics.

Part I

RNA FOLDING

This first part of our thesis provides a literature review on existing computational tools addressing the prediction of RNA secondary structure, and it presents our proposed tool `RAFFT`. Chapter 3 contains figures and ideas that have previously appeared in our publication:

- [138] Vaitea Opuu, **Nono SC Merleau**,Vincent Messow and Matteo Smerlak(2021). *RAFFT: Efficient prediction of RNA folding pathways using the fast Fourier transform.* In: *bioRxiv* (Submitted and accepted) (PLoS Comp. Biol.)

2

INTRODUCTION TO RNA FOLDING

We provided some motivations for studying ncRNAs and introduced their bioinformatic concepts in the introduction. We also highlighted the relationship between the structure of ncRNAs and their functions. The functions of ncRNAs and their lengths usually distinguish them, and several ncRNA classes were presented. Identifying the ncRNA functions is challenging, though there is a widespread expectation that their functions are largely determined by their structures. The process of determining the RNA structure is often termed RNA folding. Experimental methods that determine the secondary structure of such molecules are usually expensive. Many computational methods have been developed in the last decades as alternatives. This chapter overviews computational methods for predicting RNA secondary structures. Two techniques will be reviewed: statistical approaches such as machine learning and score-based methods.

2.1 STABILITY AND PREDICTION OF RNA SECONDARY STRUCTURES

The mapping from RNA sequences to their corresponding secondary structure defines the folding of RNA molecules. RNA folding is, therefore, a process by which a linear RNA sequence acquires a secondary structure through intra-molecular interactions. The nature of those interactions defines the thermodynamic stability of the secondary structure. Throughout this dissertation, we will denote the thermodynamic stability of a structure σ by ΔG_σ, which is the free energy difference with respect to the completely unfolded state. This section provides an intuition on how the *free energy* of an RNA secondary structure is computed based on the definitions and concepts introduced in Chapter 1. Furthermore, it introduces the problem of RNA secondary structure prediction and an overview of existing techniques.

In predicting biologically relevant structures, most computational methods search for structures that minimize the free energy function ΔG (i.e. the MFE structure). Therefore, the prerequisite to efficiently computing the MFE secondary structure is the computation of the free energy for any given secondary structure

$\mathcal{S}$. The calculation of the RNA structure free energies starts by decomposing each structure into components called loops (See Definition 4). The loop decomposition allows building the basis of the standard energy model for RNA secondary structures called the nearest neighbour (NN) model [198]. The total free energy of a secondary structure is assumed to be a sum over its constituent loops according to the additivity principle [34] (see Definition 5). Therefore, this structure decomposition allows an efficient dynamic programming (DP) algorithm to determine the MFE pseudoknot-free structure of a sequence ϕ in the structure space Σ_ϕ.

The DP is a computer programming method developed by Richard Bellman in the early 1950s [10], and it has found applications in various fields, including the RNA secondary structure prediction. It consists of simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner. When sub-problems can be nested recursively inside larger problems so that DP methods are applicable, then there is a relation between the value of the larger problem instance and the values of the sub-problems.

For example, let us consider the definition of secondary structure $\mathcal{S}$ introduced in the previous chapter (Definition 2) and its string representation σ. When considering a substructure $\sigma[i:j]$ within the sequence interval $\phi[i:j]$, there are only two alternatives to how position i may contribute to $\sigma[i:j]$. Either i does not pair with any other position, or it pairs with another nucleotide k with $i < k \leq j$. In the first situation, $\sigma[i:j]$ consists of the base pairs in the subsequence $\sigma[i+1:j]$ only. The formation of a base pair (i,k), however, subdivides the structure into two parts, one enclosed by (i,k), namely $\sigma[i+1:k-1]$, and the other one, $\sigma[k+1:j]$. Thus, $\mathcal{S} = proc\{\sigma[i+1:k-1] \cup \sigma[k+1:j]\} \cup \{(i,k)\}$, where the *proc* is the recursive procedure. Since condition (3) of definition 2 ensures that the position (i,j) can not contain base pairs that cross (i,k) (or at least in the pseudoknot-free situation), the two shorter substructures $\sigma[i+1:k-1]$ and $\sigma[k+1:j]$ can be treated independently for a large variety of purposes.

This observation has led to a recursive decomposition scheme for RNA secondary structures, which is the basis of the large variety of DP approaches that solve RNA secondary structure prediction problems. The first DP algorithm was then proposed by Nussinov and Jacobson [136] to find the structure with the maximum base pairs. A few years after, Zucker and Stieger [228] extended Nussinov's algorithm to a more realistic scoring model

based on free energy, the NN model. Almost all score-based methods rely on the same DP algorithm, but the decomposition scheme and the scoring model could differ from one to another. When predicting structures with non-canonical base pairs, some other scoring schemes are used, such as nucleotide cyclic motifs score system [28, 140, 174] or equilibrium partition function [177].

In addition to score-based methods, we have comparative sequence analysis methods, which are the most computationally accurate for determining RNA secondary structures [72, 119]. Using the set of homologous structures, the comparative method allows finding base pairs that covary to maintain WC and wobble bases of a given sequence ϕ [73]. The first comparative method predicting a common secondary structure conserved in the given homologous sequence set was developed by Han and Kim in the early nineteenth century, and it was based on comparative phylogenetic analysis.

When neglecting the special base pairs (or pseudoknots) and the weak interactions, the running time of both approaches (sore-based and comparative analysis) is usually $O(L^3)$ (Where L is the RNA sequence length) and thus prohibitingly slow for longer sequences.. Many other comparative analysis methods and variations of score-based methods were also proposed to improve computational time. More recently, a heuristic method such as `LinearFold` allows achieving good RNA folding performance in a linear time ($O(L)$).

When pseudoknots are considered, the loop decomposition of a secondary structure and the energy rules break down. Although we can assign reasonable free energies to the helices in a pseudoknot and even to possible coaxial stacking between them, it is impossible to estimate the effects of the new kinds of loops created. Base triples pose an even greater challenge because the exact nature of the triple cannot be predicted in advance, and even if it could, we have no data for assigning free energies. Nevertheless, there are existing techniques that approximate the energies of pseudoknot loops and allow the dynamic programming technique to tackle the RNA folding with pseudoknots. However, the time complexity still remains the main problem. Using the DP technique for the pseudoknot structure prediction, the time complexity goes up to $O(L^6)$ for the exact prediction. But for heuristic methods such as `IPKnot` [162] and `Hotknots` [147], the running time can be reduced down to $O(L^4)$.

Despite the advanced development of computational tools for RNA folding, it's challenging to understand the folding mecha-

nism fully. In contrast to score-based and comparative analysis methods, machine learning methods are data-driven methods that require no knowledge of the folding mechanism. Nevertheless, the requirement of ML-based methods is a large amount of training data on which they can learn. In the last few decades, ML methods have been used for many aspects of RNA secondary structure prediction methods to improve the prediction performance and overcome the limitations of existing methods. However, they did not replace the mainstream score-based methods with respect to accuracy and generalization. In addition to some overfitting concerns, ML-based methods cannot give dynamic information on the RNA folding process since little data are available on structural dynamics. In addition, the training data used in ML-based methods are mostly obtained through phylogenetic analyses. Consequently, their prediction may be biased due to the *in vivo* third elements. The following subsections provide a detailed description of some of the recent ML-based and score-based tools for secondary structure prediction.

In sum, computational methods usually consider the MFE secondary structure as the most biologically relevant one. Predicting the MFE structure consists of solving a free energy optimization problem in the case the scoring function is the free energy. Existing methods for RNA secondary structures prediction can be clustered into three main categories: the scored-based, comparative sequence analysis and ML methods. The DP technique is one of the most widely used score-based methods, but they are usually less accurate than the comparative methods. In contrast, ML methods are more recent and still under intensive improvements. The following section will overview some existing tools and highlight their limitations.

2.1.1 *MFE prediction tools for pseudoknot-free RNA sequences using a score-base method*

The score-based methods often assume that the native or biological RNA structure is the one that minimizes/maximizes the overall total score, depending on the hypotheses made on the RNA folding mechanism. In the pseudoknot-free MFE prediction, where the special and weak interactions are neglected, the folding problem is less complex, and the scoring model is simply the free energy. Hence, the issue of RNA secondary structure prediction becomes an optimization problem that aims at finding the

best-scoring structure $\mathcal{S}^{MFE}$ by minimizing a scoring function ΔG.

$$\mathcal{S}^{MFE} = \mathbf{argmin}_{\mathcal{S}\in\Sigma_\phi} \Delta G(\mathcal{S}, \phi) \tag{2.1}$$

Where Σ_ϕ is the set of all possible pseudo-knot free secondary structures for the sequence ϕ of length L and, $\Delta G(\mathcal{S}, \phi)$ the free energy of the structure $\mathcal{S}$ evaluated for the sequence ϕ.

Since each possible structure can be uniquely and recursively decomposed into smaller components (or loops) with independent free energy contributions, the DP is best suited for most of the following tools presented here.

- `Unfold` [227, 228]: It is the successor of the original `mfold` program which was the first realistic implementation of the DP for secondary structure predictions with a score based on the loop energy parameters and a worse case time complexity of $O(L^3)$. The initial version was an improvement of the simplest DP for secondary structure prediction known as the *maximum circular matching problem*. The authors demonstrated that the loop-based energy model is also amenable to the same algorithmic ideas. With McCaskill's algorithm [127], for computing the partition function of the equilibrium ensemble of RNA molecules, more efficient implementations of the initial program with accurate thermodynamic modelling have been provided. The latest implementation is known as `Unfold`.

- `RNAStructure` [125, 148]: The software first appeared in 1998 as a reimplementation of the program `mfold` with improved thermodynamic parameters. In its initial version, four major changes were made in `mfold`: (1) an improvement on the methods for forcing base pairs; (2) a filter that removed isolated WC or wobble base pairs has been added; (3) the energy parameter for interior, internal and hairpin loops were incorporated; (4) a new model for coaxial stacking of helices. It predicts the lowest free energy structure and a set of low energy structures. The new implementation also provided a user-friendly graphical interface for Windows operating system. Subsequently, the first implementation was extended to include biomolecular folding; an algorithm that finds low free energy structures common to two sequences; the partition function algorithm and all free

energy structures, and the constraints with enzymatic data and chemical mapping data. The recent version includes the partition function computation for secondary structures common to two sequences and can perform stochastic sampling of common structures [75]. Additionally, it contains `MaxExpect`, which finds maximum expected accuracy structures [116], and a method for removal of pseudoknots, leaving behind the lowest free energy pseudoknot-free structure.

- `RNAfold` [80, 112]: It is one of the most used and efficient folding tools. It computes the MFE secondary structure using an efficient DP scheme and backtraces an optimum structure. It also allows computing the partition function using McCaskill's algorithm, the matrix of base pairing probabilities, and the centroid structure. It is part of the `ViennaRNA Package`. Since its first version, it aims at suggesting an efficient implementation of Zucker's algorithm with more flexibility on the folding constraints. Many other versions have been released, including a graphics processing unit (GPU) implementation. The latest stable release of the `ViennaRNA Package` is Version 2.5.0.

- `LinearFold` [84]: For many decades, the DP techniques have been the most accurate and fast at predicting pseudoknot-free structure for short input RNA sequences. But for long sequences, the prediction remains challenging because of the computational time and the lack of accurate thermodynamic energy parameters. In contrast to traditional DP methods which are often bottom-up, `LinearFold` is a left-to-right DP. The left-to-right DP consists of scanning the input RNA sequence ϕ from left to right, maintaining a *stack* along the way and performing one of the three actions (*push*, *skip* or *pop*). The *stack* consists of a list of unpaired opening bracket positions and at each position $j = 1 \dots L$, the three actions consist respectively of 1)push: opening a bracket at position j, 2) skip: unpaired nucleotide at position j and 3) pop: closing the bracket at position j. Initially, `LinearFold`'s computational time was similar to the classical DP ($O(L^3)$) because of the *pop* action that involves three free indices (i.e. unpaired positions). But using a beam search heuristic, the time complexity was then reduced to $O(Lb \log b)$, where b is the beam size. The beam search is a popular heuristic technique used in computational linguistics. This

technique allows keeping only the top b highest-scoring (or low energy) states for each prefix of the input sequences.

Although the score-based approaches for RNA structure prediction often offer good accuracy and generalization, the non-availability of the thermodynamic energy parameters for specific loops of extended sizes presents the main challenge for predicting long sequences (i.e. $L \geq 1,000$ nucleotides). Early ML-based methods aim to improve the energy parameters by learning the underlying folding patterns from a more considerable amount of training data. In the next section of this chapter, we will present some of the recent improvements in structure prediction using ML-based methods.

2.1.2 *ML-based methods*

In the previous section, we reviewed the score-based RNA secondary structure prediction methods in general and four tools in particular, i.e. `Unfold`, `RNAstructure`, `RNAfold`, and `LinearFold`. These methods are thermodynamic methods that usually rely on experimentally energy parameters. For example, most experimental energy parameters are available only for short RNA sequences (e.g. with a length of fewer than 200 nucleotides). This limitation significantly degrades the prediction performance of thermodynamic methods for long RNA sequences. In an attempt to improve these methods, ML methods have been proposed. This section presents an overview of existing ML methods, especially those used in Chapter 3 for benchmark comparison with our proposed method.

The ML-based methods for RNA secondary structure prediction can generally be classified into three categories according to ML's subprocess, i.e., score scheme based on ML, preprocessing and postprocessing based on ML, and prediction process on ML. All the ML-based methods in these three categories trained their models in a supervised way [225].

When using a scoring scheme based on ML, the parameter estimation in the scoring scheme is first optimized using an ML model. The estimated parameters are then used to evaluate the scores of possible conformations. Difference scoring schemes can be refined by using that approach: the free energy parameters, weights, and probabilities. The free energy parameter-refining is the most popular because several thermodynamic parameters of the NN model have to be based on a large number of optimal

melting experiments and the experiments are time and labour-consuming. In fact, not all free energy changes in structural elements can be experimentally measured because of technical difficulties. Instead of refining the free energy parameters, some ML-based approaches scream through existing data of RNA structures to extract weights that consist of different features of RNA structure elements. These weights can be used as a scoring function for DP techniques. The advantage of such a scoring function is that it decouples structure prediction and energy estimation. However, learned weights have no explanations because of the ML black box.

Another alternative for predicting RNA structures is the stochastic context-free grammar (SCFG) [41, 102, 103, 151, 157, 214]. SCFGs allow building grammar rules and induce a join probability distribution over possible RNA structure for a given sequence ϕ. In addition, the SCFG models specify probability parameters for each production rule in the grammar, which allow assigning a probability to each sequence generated by the grammar. These probability parameters are learned from datasets of RNA sequences associated with known secondary structures without carrying any external laboratory experiments [41].

Besides the ML-based methods that focus on refining the folding parameters, there are preprocessing and post-processing based on ML [77, 83, 226] and direct predicting process based on ML [111, 184, 187]. Preprocessing and postprocessing models allow for choosing the appropriate prediction method or set of prediction parameter sets and provide a means of determining the most likely structures among the possible outcomes that are useful for decision. The preprocessing and postprocessing ML tools are often based on a support vector machine (SVM).

Finally, it is possible to use ML techniques to predict RNA secondary structure directly or combine it with other algorithms in an end-to-end fashion. Below are some of the most used and recent ML-based tools for RNA secondary structure prediction.

- `ContraFold`[38]: Using the so-called probabilistic model, the conditional log-linear model (CLLM), `ContraFold` appeared for the first time in early 2006. It was the first probabilistic prediction tool outperforming the existing tools, including thermodynamic tools such as `RNAfold` and `mfold`. The CLLM is a flexible class of probabilistic models that generalizes upon SCFGs, using discriminative training and feature-rich scoring. The tool implements a CLLM incorporating most of the features found in typical thermodynamic

models allowing the tool to achieve the highest single sequence prediction accuracy to date when compared with the currently available probabilistic models.

- `ContextFold` [220]: In contrast to `ContraFold`, `ContextFold` utilizes a weighted approach based on ML. In particular, it uses a discriminative structured-prediction learning framework combined with an online learning algorithm. `ContextFold` uses a large training dataset of RNA sequences annotated with their corresponding structures to obtain an ML model made of 70,000 free parameters, which has several orders of magnitudes compared to traditional models (i.e. thermodynamic free energy parameters). At its first apparition, `ContextFold`'s model succeeded at the error reduction of about 50%. Still, some overfitting concerns have been reported when using the tool, especially for predicting structures with large unpaired regions.

- `Mxfold2` [160]: It is one of the most recent ML-based tools for predicting the secondary structure of RNA molecules. Its particularity is the ML technique used, a ML it also belongs to the weighted approach based on ML since the resulting model of a deep neural network (DNN) is a set of weight parameters. `MxFold2`'s DNN uses the max-margin framework with thermodynamic regularization. It made the folding scores predicted by `Mxfold2` and the free energy calculated by the thermodynamic parameters as close as possible. This method has shown robust prediction on both sequences and families of natural RNAs, suggesting that the weighted ML approaches can compensate for the gaps in the thermodynamic parameter approaches.

Although ML methods provide substantial improvements compared to traditional methods such as thermodynamic and comparative sequence analysis [161, 175], they often lack physical principles (training data are mostly obtained through phylogenetic analyses) and present some over-fitting concerns [152]. In addition to the over-fitting problems partially due to few data availability, ML methods do not provide dynamic information on RNA folding for the same reason. In Chapter 3, we will introduce our approach that aims at predicting an ensemble structure, which allows us to derive some dynamic information and contrasts the methods previously presented.

2.1.3 *Prediction tools for pseudoknotted RNA sequences*

In the introduction, we have provided the importance of pseudoknot interaction in realizing biological functions, and different pseudoknot patterns have been reviewed. This section introduces a couple of tools for predicting RNA pseudoknotted structures that will be used in the benchmark results presented in Chapter 5.

Folding RNA sequences with pseudoknotted interactions is computationally more expensive than a pseudoknot-free target. Specifically, the time complexity of the pseudoknot-free secondary structure prediction is $O(L^3)$ when using dynamic programming approaches such as `RNAfold`, or less with heuristic folding methods (e.g. $O(L)$ for `LinearFold` and $O(L^2 \log L)$). By contrast, when considering a special class of pseudoknots, the time complexity of folding goes up to $O(L^6)$ for an exact thermodynamic prediction using a dynamic programming approach such as [150]. When Using heuristic methods, the time complexity slows down to $O(L^4)$ (e.g. tools such as `IPknot` and `HotKnots`) or $O(L^3)$ for tool such as `HFold`.

- `pKiss` [90]: The program `pKiss` appears the first time in 2014 as an updated version of the program `pknotsRG`[144] which is a module of the RNA abstract shapes analysis `RNAshapes` [90]. Initially, the program `pknotsRG` was built for the prediction of some special class of pseudoknots (unknotted structures and H-type pseudoknots). Later on, it was extended to predict RNA structures that exhibit kissing hairpin motifs in an arbitrarily nested fashion, requiring $O(L^4)$ time. In addition to predicting the kissing hairpin motifs, `pKiss` also provides new features such as shape analysis, computation of probabilities, different folding strategies and different dangling base models.

- `IPknot [162]`: it was first introduced in a paper by Kengo and his collaborators in 2011 as a novel computational tool for predicting RNA secondary structure with pseudoknots using integer programming technique. `IPknot` uses the maximum expected accuracy (MEA) as a scoring function, and the maximizing expected accuracy problem is solved using integer programming with threshold cut. `IPknot` decomposes a pseudoknotted structure into a set of pseudoknot-free substructures and approximates a base-pairing probability distribution that considers pseudoknots, leading to the capability of modelling a comprehensive class of

pseudoknots and running quite fast. In addition to single sequence analysis, `IPknot` can also predict the consensus secondary structure with pseudoknots when a multiple sequence alignment is given.

- `HotKnots` [147]. In contrast to the previously mentioned tools, `HotKnots` implements a heuristic algorithm based on the simple idea of iteratively forming stable stems. The algorithm explores many alternative secondary structures using a free energy minimization for pseudoknot-free secondary structures. Several other additions of a single substructure are considered for each structure formed at each step, resulting in a tree of candidate structures. The criterion for determining which substructures to add to partially formed structures at successive levels of the tree was also new. Similar to previous algorithms, energetically favourable substructures called *hotspots* are found by a call to Zuker's algorithm, with the constraint that no base already paired may be in the structure.

Despite the higher computational complexity of pseudoknots, it is still important to account for them as they occur in natural RNA and are relevant for RNA function. We have reviewed three mainly used RNA secondary prediction tools (`pKiss`, `IPknot`, `HotKnots`) that support the two pseudoknot patterns (i.e. the H-type and K-type) considered in Chapter 5. In addition to the computational complexity, existing methods lack experimentally measured energy parameters for pseudoknot interactions. Therefore, they mostly rely or do not on approximated energy parameters, which may influence the predictions. Only `IPknot` and `HotKnots` will be used among these tools when designing pseudoknotted RNA structures. `HotKnots` predicts the free energy of pseudoknotted structure based on recently updated energy parameters, whereas `IPknot` does not.

So far, we have presented tools that predict a single stable and static RNA secondary structure for a given RNA sequence, including pseudoknots or not. More often than not, the ncRNA functions are associated with the RNAs' ability to undergo specific conformational changes, as is the case for riboswitches. The function of an RNA molecule thus is usually poorly described by its ground state structure and instead has to be studied as a dynamic ensemble of structures [35, 137]. The following section will review some computational methods that address the folding dynamics of RNA molecules.

2.2 RNA KINETICS

The previous section introduced how pseudoknot-free secondary structures with their thermodynamic properties can be predicted. It also introduced some statistical methods that do not only rely on the thermodynamic principle but training data obtained from phylogenetic analysis, mainly the ML methods. However, the methods used for predictions do not tell us anything about how the structures change over time and how they are related to each other. This section discusses the folding dynamics of RNA molecules.

The folding of RNA molecules is remarkably more complex. It is a result of the delicate balance between multiple factors: the chain entropy, ion-mediated electrostatic interactions and solvation effect, base pairing and stacking, and other non-canonical interactions [22]. It is a dynamic process governed by a constant formation or dissolving of base pairs. In other terms, the RNA molecule navigates its structure space by following a free energy landscape. Here, the free energy landscape is a high-dimensional space of all possible secondary structures (Σ_ϕ) weighted by their free energy ΔG.

As usually done, the kinetics is modelled as a continuous-time Markov chain [114], where populations of structure evolve according to transition rates. In this context, an Arrhenius formulation is commonly used to derive elementary transition from state i to state j; where $\Delta G^{\ddagger}_{i\to j}$ is the activation barrier separating i from j, and $\beta = 1/k_B T$ is the inverse thermal energy (mol/kcal).

$$k_{i\to j} = k_0 \exp(-\beta \Delta G^{\ddagger}_{i\to j}) \tag{2.2}$$

Here k_0 is the actual rate constant, solvent-dependent. Three rate models describing elementary steps in the structure space are often used to study RNA folding dynamics:

1. The base stack model [221–223]: it uses base stacks as elementary kinetic move. A move consists of an addition or a breaking of a base stack with $\Delta G^{\ddagger}_{i\to j}$ equal to the change in the entropic free energy $T\Delta S$ and the enthalpy ΔH, respectively.

2. The base pair model [24, 53]: it uses base pair as elementary kinetic steps which gives the finest resolution, but at the cost of computation time. Here $\Delta G^{\ddagger}_{i\to j} = \Delta G/2$ where ΔG is

the energy change from state i to state j or $\Delta G^{\ddagger}_{i\to j} = \Delta G$ for $\Delta G \geq 0$.

3. The helix stem model [87, 121]: the elementary move is the creation or deletion of a helix stem. It provides a coarse-grained description of the dynamics where free energy changes ($\Delta G^{\ddagger}_{i\to j}$) due to stem formation guiding the folding process.

The different rate models can lead to different folding pathways. The key factor that distinguishes the different rate models is whether the barrier is determined by $(\Delta H, \Delta S)$ or by ΔG. The $(\Delta H, \Delta S)$ values for different RNA base stacks show well-separated discrete hierarchies, whereas the ΔG values show no such large separation. For two typical base stacks, 5′AU-AU3′ and 5′UC-GA3′, the difference $\Delta(\Delta H_{stact}, \Delta S_{stack}) = (7.4\ \text{kcal/mol}, 20\ \text{kcal/mol})$ is much larger than the difference $\Delta(\Delta G_{stack}) = 1.4\text{kcal/mol}$[168]. Because of this fact, different models can give different folding kinetics.

Depending on the rate model used, the following master-equation describe the population kinetics $p_i(t)$ for the i^{th} state ($i = 1 \ldots \Omega$, where Ω is the total number of chain conformations).

$$\frac{\mathrm{d}p_i(t)}{\mathrm{d}t} = \sum_{j\in\Omega} k_{j\to i} p_j(t) - k_{i\to j} p_i(t), \tag{2.3}$$

where $k_{j\to i}$ and $k_{i\to j}$ are the rate constants for the respective transitions. The equivalent matrix form of Equation 2.3 is given by:

$$\frac{\mathrm{d}\mathbf{p}(t)}{\mathrm{d}t} = \mathbf{M}.\mathbf{p}, \tag{2.4}$$

where $\mathbf{p} = (p_i, \ldots p_\Omega)$ is a column vector representing the frequency of structure at state $(i, \ldots, \Omega)$ and, $\mathbf{M}$ is the rate matrix defined as:

$$\mathbf{M}_{ij} = \begin{cases} k_{i\to j}, & \text{if } i \neq j \\ -\sum_{j\neq i} k_{ij}, & \text{if } i = j \end{cases} \tag{2.5}$$

For a given initial folding condition $p_i(0)$, the Equation 2.4 is solvable by diagonalizing the rate matrix $\mathbf{M}$ and, the solution is the population kinetics $\mathbf{p}(t)$ for $t > 0$ is given by:

$$\mathbf{p}(t) = \sum_{m=1}^{\Omega} C_m \mathbf{n}_m \exp -\lambda_m t \tag{2.6}$$

where $-\lambda$ and $\mathbf{n}_m$ are the m^{th} eigenvalue and eigenvector of the rate matrix $\mathbf{M}$, and C_m is the coefficient that is dependent on the initial condition. The eigenvalue spectrum gives the rates of the kinetic modes of the system.

Simulating the RNA dynamics using Equation 2.3 has some limitations. The solution to the master-equation given by Equation 2.6 can only give ensemble-average macroscopic kinetics and cannot give detailed information about the microscopic pathways [224]. Moreover, the number of structures (Ω) increases rapidly with the RNA sequence length L. Therefore, the master equation is often limited to short RNA sequences. Because of these limitations, kinetics-cluster methods are alternatively used. The basic idea of the kinetic-cluster method is to classify the large structural ensemble into a much-reduced system of clusters (of macrostates) such that the inter-cluster transitions can represent the overall kinetics. Although both the master-equation and the kinetic-cluster methods can predict the macroscopic kinetics, the kinetic-cluster approach has the unique advantage of providing direct information on the microscopic pathway statistics from the inter-cluster transitions [224]. Both approaches are based on the complete conformational ensemble. An alternative approach, implemented in `kinwalker` [62], used the observation that folded intermediates are generally locally optimal conformations. Like thermodynamic methods for static RNA secondary structure prediction, experimental studies usually play an essential role in guiding computational methods in studying RNA folding dynamics. Several recent observations are discussed in the following paragraph.

In folding experiments, Pan and coworkers observed two kinds of pathways in the free energy landscape of a natural ribozyme [139]. Firstly, the investigations revealed fast-folding pathways, in which a subpopulation of RNAs folded rapidly into the native state. However, the second population quickly reached metastable misfolded states, then slowly folded into the native structure. In some cases, these metastable states are functional. These phenomena are direct consequences of the rugged nature of the RNA folding landscape [180].

The experiments performed by Russell and coworkers also revealed the presence of multiple deep channels separated by high energy barriers on the folding landscape, leading to fast and slow folding pathways [156]. The formal description of the above mechanism, called the kinetic partitioning mechanism, was first introduced by Guo and Thirumalai in the context of

protein folding [69]. These metastable conformations constitute competing attraction basins in the free energy landscape where RNA molecules are temporarily trapped. However, *in vivo*, folding into the native states can be promoted by molecular chaperones [20], which means that the active structure depends on factors other than the sequence. This may raise some discrepancies when comparing thermodynamic modelling to actual data.

The experimental verification of the rate model is also a challenge because the microscopic elementary processes are hidden in the ensemble averages of the measured kinetics. Many researchers believe that single-molecule experiments may provide a discerning measure with careful extrapolation to the force-free case. All atom-simulations with a reliable force field and sampling method are highly valuable for providing detailed atomistic configurations for the transition state [22]. Alternatively, systematic theory-experiment tests as done in [224] for designed sequences can also provide critical assessment for the different rate models.

In sum, studying the folding of RNA molecules as a dynamic ensemble of structures is of central importance in describing their functions, and experimental observations often guide the computational methods. Some of the recent experimental observations have been reviewed in this section. Among them, the kinetic partitioning mechanism is of interest in this work. It revealed the presence of multiple deep channels separated by high-energy barriers on the folding landscape, which leads to fast and slow folding pathways. The folding tool we suggest in Chapter 3 is inspired by this mechanism and predicts fat RNA folding pathways. The predicted pathways, therefore, allow us to derive dynamic information on RNA folding.

2.3 CONCLUSION

In this chapter, we have presented the RNA folding in two main steps: (1) the prediction of the secondary structure of RNA, which represents the static part of the folding process; (2) the RNA kinetics, which aim at modelling the dynamics of the folding. RNA secondary structure prediction was introduced as an optimization problem, and a review of existing methods and tools was presented. Of particular importance in this thesis's context is that existing tools for predicting RNA secondary structures often present some limits in computational time for longer RNA sequences. Mainly the existing tools do not give dynamical information, as few data are available on structural dynamics. Sim-

ulating the folding kinetics of long RNA molecules is also of an essential limit because it requires a full enumeration of the structural space in most cases. In the next chapter, we will present our thesis's first result, which aims to predict RNA folding pathways efficiently using the FFT. The predicted pathways allow us to derive energetically suboptimal structures from which we model the RNA folding kinetics with fewer secondary structures.

3

RAFFT: EFFICIENT PREDICTION OF FAST-FOLDING PATHWAYS OF RNAS

This chapter introduces a novel heuristic algorithm to predict an ensemble of metastable RNA secondary structures for a given sequence ϕ. The algorithm is inspired by the kinetic partitioning mechanism, by which molecules follow alternative folding pathways to their native structure, some much faster than others. Similarly, our algorithm RAFFT generates an ensemble of concurrent folding pathways ending in multiple metastable structures for each given sequence. We then use the ensemble structures as finite ensemble states in which the RNA sequence can be at a given time, and the energy difference from one state to another is then used to derive a stem rate model. Therefore, our algorithm also acts as a folding kinetic ansatz. Much of the material in this chapter has been previously described in [138].

3.1 MATERIAL AND METHODS

The computational time is one of the challenges for the existing tool in folding long RNA molecules. The method we present in this work aims to improve the existing RNA folding tools reviewed in Chapter 2. It is based on the FFT and inspired by the kinetic partitioning mechanism. As presented in Chapter 1, the FFT allows reducing the computational time of the correlation between two sequences. We use the same ideal in the context of this work to faster predict RNA folding pathways by analyzing high correlation positional lag between an RNA sequence and its complementary copy, especially for longer sequences. We, therefore, derive a kinetics ansatz from the structural ensemble of the predicted folding paths. This section describes our RNA pathways prediction method and the kinetics ansatz derived from the predicted structural ensemble. In addition, it provides a description of the benchmark dataset used to assess our method performance and the comparison protocols.

3.1.1 *RAFFT's algorithm description*

RAFFT starts from a sequence of nucleotides $\phi = (\phi_1 \dots \phi_L)$ of length L, and its associated unfolded structure σ. We first create a numerical representation of ϕ where each nucleotide is replaced by a unit vector of 4 components:

$$\mathrm{A} \to \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \mathrm{U} \to \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \mathrm{C} \to \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \mathrm{G} \to \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \tag{3.1}$$

This encoding gives us a $(4 \times L)$-matrix we call X, where each row corresponds to a nucleotide as shown below:

$$X = \begin{pmatrix} X^{\mathrm{A}} \\ X^{\mathrm{C}} \\ X^{\mathrm{G}} \\ X^{\mathrm{U}} \end{pmatrix} = \begin{pmatrix} X^{\mathrm{A}}(1) & X^{\mathrm{A}}(2) & \dots & X^{\mathrm{A}}(L) \\ X^{\mathrm{C}}(1) & X^{\mathrm{C}}(2) & \dots & X^{\mathrm{C}}(L) \\ X^{\mathrm{G}}(1) & X^{\mathrm{G}}(2) & \dots & X^{\mathrm{G}}(L) \\ X^{\mathrm{U}}(1) & X^{\mathrm{U}}(2) & \dots & X^{\mathrm{U}}(L) \end{pmatrix} \tag{3.2}$$

For example, $X^{\mathrm{A}}(i) = 1$ if $\phi_i = \mathrm{A}$. Next, we create a second copy $\bar{\phi} = (\bar{\phi_L} \dots \bar{\phi_1})$ for which we reversed the sequence order. Then, each nucleotide of $\bar{\phi}$ is replaced by one of the following unit vectors:

$$\bar{\mathrm{A}} \to \begin{pmatrix} 0 \\ 0 \\ 0 \\ w_{\mathrm{AU}} \end{pmatrix}, \bar{\mathrm{U}} \to \begin{pmatrix} w_{\mathrm{AU}} \\ w_{\mathrm{GU}} \\ 0 \\ 0 \end{pmatrix}, \bar{\mathrm{C}} \to \begin{pmatrix} 0 \\ 0 \\ w_{\mathrm{GC}} \\ 0 \end{pmatrix}, \bar{\mathrm{G}} \to \begin{pmatrix} 0 \\ w_{\mathrm{GC}} \\ 0 \\ w_{\mathrm{GU}} \end{pmatrix}. \tag{3.3}$$

$\bar{\mathrm{A}}$ (respectively $\bar{\mathrm{U}}, \bar{\mathrm{U}}, \bar{\mathrm{G}}$) is the complementary of A (respectively U, C, G). w_{AU}, w_{GC}, w_{GU} represent the weights associated with each canonical base pair, and they are chosen empirically. We call this complementary copy $\bar{X}$, the mirror of X.

To search for stems, we use the complementary relation between X and $\bar{X}$ with the correlation function $\mathrm{cor}(k)$. This correlation is defined as the sum of individual X and $\bar{X}$ row correlations:

$$\mathrm{cor}(k) = \sum_{\alpha \in \{\mathrm{A,U,C,G}\}} c_{X^\alpha, \bar{X}^\alpha}(k), \tag{3.4}$$

where a row correlation between X and $\bar{X}$ is given by:

$$c_{X^\alpha, \bar{X}^\alpha}(k) = \sum_{\substack{1 \le i \le L \\ 1 \le i+k \le L}} \frac{X^\alpha(i) \bar{X}^\alpha(i+k)}{\min(k, 2L-k)}. \tag{3.5}$$

For each $\alpha \in \{A, U, C, G\}$, $X^{\alpha}(i) \times \bar{X}^{\alpha}(i+k)$ is non zero if sites i and $i+k$ can form a base pair, and will have the value of the chosen weight as described above. If all the weights are set to 1, $\mathrm{cor}(k)$ gives the frequency of base pairs for a positional lag k. Although the correlation naively requires $O(L^2)$ operations, it can take advantage of the FFT which reduces its complexity to $O(L \log(L))$.

Large $\mathrm{cor}(k)$ values between the two copies indicate positional lags k where the frequency of base pairs is likely to be high. However, this does not allow to determine the exact stem positions. Hence, we use a sliding window strategy to search for the largest stem within the positional lag (since the copies are symmetrical, we only need to slide over one-half of the positional lag). Once the largest stem is identified, we compute the free energy change associated with the formation of that stem. Next, we perform the same search for the n highest correlation values, which gives us n potential stems. Then, we define as the current structure the stem with the lowest free energy. Here, free energies were computed using Turner2004 energy parameters through ViennaRNA package API [112].

We are now left with two independent parts, the interior and the exterior of the newly formed stem. If the exterior part is composed of two fragments, they are concatenated into one. Then, we apply recursively the same procedure on the two parts independently in a *breadth-first* fashion to form new consecutive base pairs. The procedure stops when no base pair formation can improve the energy. When multiple stems can be formed in these independent fragments, we combine all of them and pick the composition with the best overall stability. If too many compositions can be formed, we restrict this to the 10^4 bests in terms of energy. Figure 3.1 shows an example of execution to illustrate the procedure.

The algorithm described so far tends to be stuck in the first local minima found along the folding trajectory. To alleviate this, we implemented a stacking procedure where the N best trajectories are stored in a stack and evolved in parallel. Like the initial version, the algorithm starts with the unfolded structure; then, the N best potential stems are stored in the first stack. From these N structures, the procedure tries to add stems in the unpaired regions left and saves the N best structures formed. Once no stem can be formed, the algorithm stops and output the structure with the best energy found among the structures stored in the last stack. This algorithm leads to the construction of a graph we call

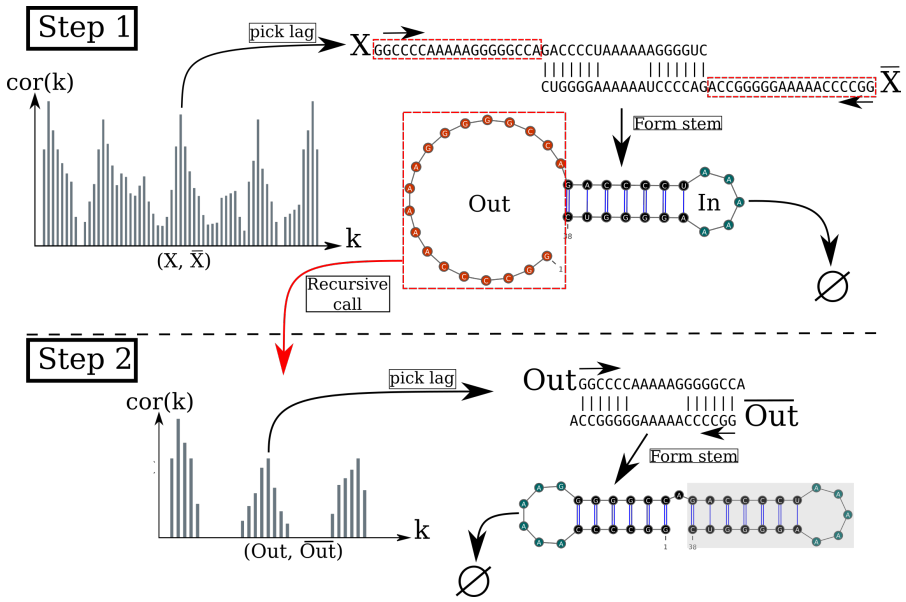


Figure 3.1: **Algorithm execution for one example sequence which requires two steps.** (Step 1) From the correlation $cor(k)$, we select one peak which corresponds to a position lag k. Then, we search for the largest stem and form it. Two fragments, "In" (the interior part of the stem) and "Out" (the exterior part of the stem), are left, but only the "Out" may contain a new stem to add. (Step 2) The procedure is called recursively on the "Out" sequence fragment only. The correlation $cor(k)$ between the "Out" fragment and its mirror is then computed and analyzing the k positional lags allows to form a new stem. Finally, no more stem can be formed on the fragment left (colored in blue), so the procedure stops.

a *fast-folding graph*. In this graph, two structures are connected if the transition from one to another corresponds to the formation of a stem or if the two structures are identical. Figure 3.2 shows an example of a *fast-folding graph* produced by RAFFT for $N = 5$.

This section presented the complete procedure implemented in our proposed tool RAFFT. The procedure resulted in an ensemble of concurrent folding pathways ending in multiple metastable secondary structures. The connections in each folding pathway are dictated by the formation of stems, resulting in an energy increase. The different folding pathways connected to the initial unfolded structure form a fast folding graph. The ensemble of secondary structures constituting the fast folding graph is then used to build our kinetics ansatz where the transitions follow the Metropolis rules, i.e. no barriers between structures. The

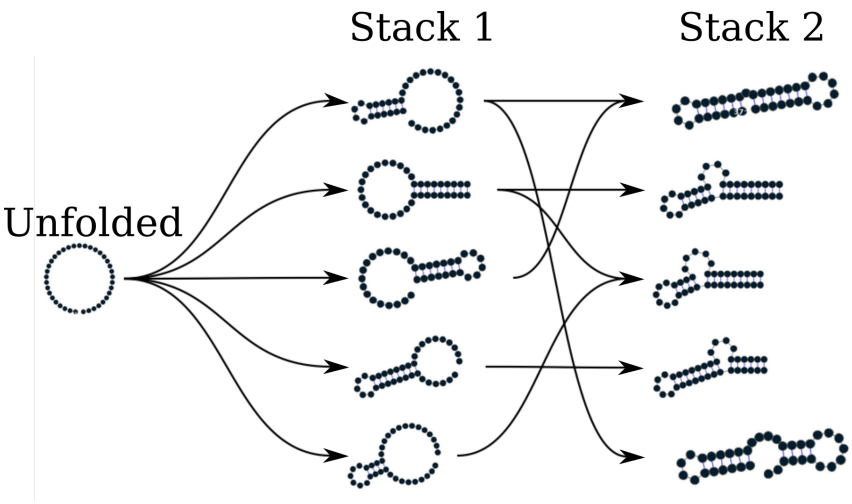


Figure 3.2: **Fast folding graph constructed using RAFFT.** In this example, the sequence is folded in two steps. The algorithm starts with the unfolded structure on the left. The $N = 5$ best stems are stored in stack 1. From stack 1, multiple stems formation are considered, but only the $N = 5$ best are stored in stack 2. Structures are ordered (from top to bottom) by energy in each stack. All secondary structure visualizations were obtained using VARNA [31].

following section provides more details on our proposed kinetics ansatz.

3.1.2 *Kinetic ansatz*

Now that the RNA pathway prediction algorithm is described, we provide the ingredients needed to extract dynamic folding information from the previously generated fast folding graph in this section.

The folding kinetic ansatz used here is derived from the fast-folding graph and allows us to model the slow processes in RNA folding. As described in Figure 3.2, transitions can occur from left to right (and right to left) but not vertically. The fast-folding graph follows the idea that parallel pathways quickly reach their endpoints; however, when the endpoints are non-native states, this ansatz allows slowly folding back into the native state [139].

Using the master-equation (See Equation 2.3), the traditional kinetic approach often starts by enumerating the whole space (or a carefully chosen subspace) of structures using `RNAsubopt`. Next, this ensemble is divided into local attraction basins separated from one another by energy barriers. This coarsening is usually

done with the tool called `barriers`. Then, following the Arrhenius formulation (See Equation 2.2) , one simulates a coarse grained kinetics between basins.

In contrast to traditional kinetics approaches, the connected structures in the `RAFFT`'s fast-folding graph are not always separated by activation barrier energies. Therefore, we computed the transition rates $k_{i\to j}$ using the Metropolis [101] formulation defined as follow

$$k_{i\to j} = \begin{cases} k_0 \times \min(1, \exp(-\beta\Delta(\Delta G_{i\to j}))), & \text{if } \sigma_i \in \mathcal{M}(\sigma_j) \\ 0, & \text{else} \end{cases}, \quad (3.6)$$

where $\Delta\Delta G_{i\to j} = \Delta G_j - \Delta G_i$ is the free energy change between structure σ_i and σ_j. Here, k_0 is a conversion constant that we set to 1 for the sake of simplicity and we initialize the population $p_i(0)$ with only unfolded structures; therefore, the trajectory represents a complete folding process. The frequency of a structure σ_i evolves according to the master Equation 2.3. Due to this approximation, we referred to our approach as a *kinetic ansatz*

In sum, based on the FFT, we constructed a method that allows generating an ensemble of secondary structures by a successive formation of stems. Using this ensemble, we derived a kinetics ansatz in which transitions between structures follow the Metropolis rules. We assess the performance of our tool by comparing its predictions to existing tools using benchmark datasets. The following section briefly describes the datasets used in this work, including the clean procedure applied to the initial datasets.

3.1.3 *Benchmark datasets.*

Measuring the performance of computational RNA folding tools can be quite a challenging task. A perfect validation procedure will require a comparison to experimental data, which in practice are not also perfect and are very expensive. In the context of this work, we perform *in silico* validation using benchmark datasets, which is a collection of native sequence structures. Because our proposed method produces kinetics and static structure predictions., we assess the performance of both tasks separately and using a different dataset. This section presents the two datasets.

To build the dataset for the folding task application, we started from the `ArchiveII` dataset derived from multiple sources [4, 9, 17, 29, 32, 61, 70, 71, 124, 158, 164, 176, 181, 183, 207, 229, 230]. We first removed all the structures with pseudoknots, since the

tools considered here do not handle these loops. Next, using the Turner2004 energy parameters, we evaluated the structures' energies and removed all the unstable structures: structures with energies $\Delta G > 0$. This dataset is composed of 2,698 sequences with their corresponding known structures. 240 sequences were found multiple times (from 2 to 8 times); 19 of them were mapped to different structures. For the sequences that appeared with different structures, we picked the structure with the lowest energy. In the end we arrived at a dataset with 2,296 sequences-structures.

For the kinetics task, there is no existing standard procedure or dataset allowing to validate or not a computational tool. However, for the validation of our kinetic ansatz, we used the CFSE RNA sequence and classic bi-stable sequence **GGCCCCUUUGGGGGCCAGACCCCUAAAGGGGUC**.

In sum, two different dataset sets are used to assess `RAFFT` performance: the first one, `Archive II` consists of 2,296 sequences-structures used for the prediction task, and one which contains two sequences, the CFSE and a bistable sequence for the kinetic study. The following section describes the benchmarking protocols for both tasks.

3.1.4 *Structure prediction protocols*

The static RNA structure prediction and the RNA kinetic performances of our proposed tool `RAFFT` are evaluated separately. This section describes the evaluation protocols for both performances and the different tool parameters used throughout.

To evaluate the structure prediction accuracy of the proposed method, we compared `RAFFT` to five recent secondary structure pseudoknot-free prediction tools. The five tools include ML-based methods (`Mxfold2 0.1.1` and `Contrafold`) and score-based methods (`RNAfold 2.4.13`, `Linearfold`, and `RNAstructure`). To compute the MFE structure for the score-based methods, we used the default parameters and the Turner2004 set of energy parameters. We also computed the ML predictions using the default parameters. Therefore, only one structure prediction per sequence for these tools was used for the statistics.

Two parameters are critical for `RAFFT`, the number of positional lags in which stems are searched, and the number of structures stored in the stack. For our computational experiments, we searched for stems in the $n = 100$ best positional lags and stored $N = 50$ structures. The correlation function $\mathrm{cor}(k)$ which

allows to choose the positional lags is computed using the weights $w_{GC} = 3$, $w_{AU} = 2$, and $w_{GU} = 1$.

To assess the performance of RAFFT, we analyzed the output in two different ways. First, we considered only the structure with the lowest energy found for each sequence. This procedure allows us to assess RAFFT performance in predicting the MFE structure. Second, we computed the accuracy of all $N = 50$ structures saved in the last stack for each sequence and displayed only the best structure in terms of accuracy. As mentioned previously in Chapter 2, the lowest energy structure found may not be the active structure. Therefore, this second assessment procedure allows us to show whether one of the pathways is biologically relevant.

We used two metrics to measure the prediction accuracy: the PPV and the sensitivity. The PPV measures the fraction of correct base pairs in the predicted structure, while the sensitivity measure the fraction of base pairs in the accepted structure that are predicted. These metrics are defined in Chapter 1 (See definitions 11, 12). To be consistent with previous studies, we computed these metrics using the `scorer` tool provided by Matthews *et al.* [122], which also provides a more flexible estimate where shifts are allowed.

Further more, we used a PCA to visualize the loop diversity in the predicted structures for each folding tool considered here. To extract the weights associated with each structure loop from the dataset, we first converted the structures into weighted coarse-grained tree representation [169]. In the tree representation, the nodes are generally labelled as E (exterior loop), I (interior loop), H (hairpin), B (bulge), S (stacks or stem-loop), M (multi-loop) and R (root node). We separately extracted the corresponding weights for each node, and the weights are summed up and then normalized. Excluding the root node, we obtained a table of 6 features and n entries. This allows us to compute a 6×6 correlation matrix that we diagonalize using the `eigen` routine implemented in the `scipy` package. For visual convenience, the structure compositions were projected onto the first two principal components (PCs).

Finally, the CFSE and a bistable RNA sequence are used to assess the kinetic performance. For each sequence, initial conditions are chosen for both `Treekin` and RAFFT to simulate the kinetic trajectories. Both kinetics are simulated using the master equation described in Chapter 2 (Equation 2.3) but with different transition rules, `Treekin` uses the Arrhenius rules whereas RAFFT uses the

Metropolis rules. The following section describes the statistical results obtained for both kinetics and structure prediction tasks.

3.2 EXPERIMENTAL RESULTS

The validation of our results is purely statistical, i.e. using statistical methods such as t-test and regression to compare different tool performance data. Based on the previously mentioned limitations of existing tools, we evaluate three main RAFFT potential improvements: the running time for the folding or pathways prediction, the quality of the predicted pathways and the RNA folding kinetics. This section discusses each of those performances in comparison to the existing tools.

3.2.1 *RAFFT's run time and scalability*

The first input of our method is a potential improvement to the CPU time of existing tools. This section focuses on analyzing RAFFT's running time compared to existing methods. Four different tools are considered: `RNAfold`, `ContraFold`, `RNAstructure` and `LinearFold`. All of them are MFE estimates implementing a DP with cubic time complexity$(O(L^3))$, except for `ContraFold` which implements a ML approach. When using the heuristic implementation of `LinearFold`, the time complexity is linear while losing the MFE estimation. We will first discuss RAFFT theoretical time complexity before comparing the empirical execution times to the existing tools.

The complexity of RAFFT's algorithm depends on the number and size of the stems formed. The main operations performed for each stem formed are: (1) the evaluation of the correlation function $\mathrm{cor}(k)$, (2) the sliding-window search for stems, and (3) the energy evaluation. We based our approximate complexity on the correlation evaluation since it is the more computationally demanding step; the other operations only contribute a multiplicative constant at most. The best case is the trivial structure composed of one large stem where the algorithm stops after evaluating the correlation on the complete sequence. At the other extreme, the worst case is one where at most $L/2$ stems of size 1

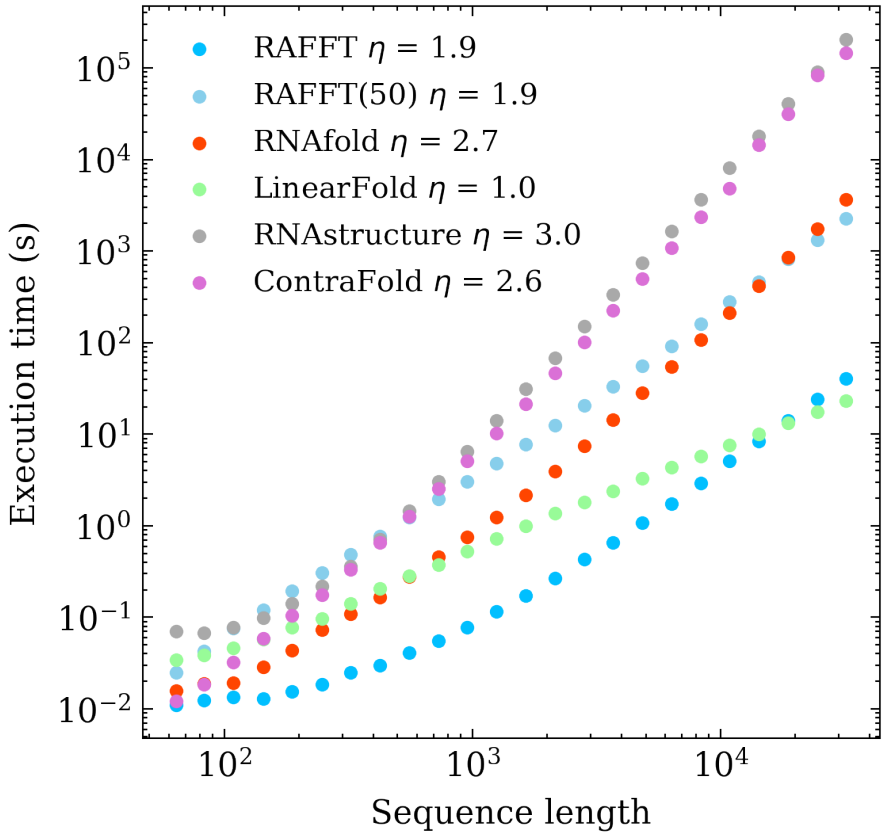
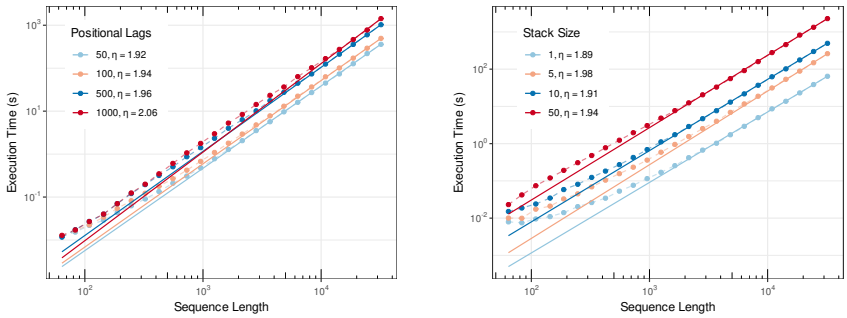


Figure 3.3: **Execution time comparisons**. For samples of 30 sequences per length, we averaged the execution times of five folding tools. The empirical time complexity $O(L^{\eta})$ where η is obtained by non-linear regression. `RAFFT` denotes the naive algorithm(with only $N = 1$ structure saved per stack), whereas `RAFFT(50)` denotes the algorithm where 50 structures can be saved per stack.

(exactly one base pair peer stems) can be formed. The approximate complexity therefore depends on

$$\sum_{i=0}^{L/2} (L-2i)\log(L-2i) \;= O(L^2 \log L) \tag{3.7}$$

We compared `RAFFT`'s execution time to the classical cubic-time algorithms represented by `CONTRAfold` (Version 2.02), `RNAstructure` (Version 2.0), `RNAfold` (Version 2.4.13) and the recent improved DP tool `LinearFold` (Version 1.0). Figure 3.3 shows the execution time of the `RUST` implementation of `RAFFT` and the four above-mentioned tools for 30 random generated sequences of various lengths. When comparing `RAFFT` implementation to the standard



(a) CPU times respect to the positional lags (n)

(b) CPU times respect to the stack size (N)

Figure 3.4: **Impact of the number of positional lags n and the stack size N on the runtime complexity.** For a corresponding length, we generated 30 random sequences, and averaged their execution times. Solid lines indicate the estimated time complexity $O(L^{\eta})$ where η is obtained with a non-linear regression on these average execution times for (A) Different number of positional lags n and (B) Different stack sizes N.

DP tools, the execution time of `RAFFT` scales slower (with an exponent ≈ 2) with the sequence length whereas the standard DP execution times are cubic. In contrast, the execution time of the improved DP implemented by `LinearFold` scales linearly with the sequence length. Only when considering a stack size of 1, that `RAFFT` execution time is lower than the one of `LinearFold` for sequence of lengths less than $L = 10^4$.

We also analyse the scalability of `RAFFT` computational time with respect to its critical parameters (the number of positional lag n and the stack size N). Figure 3.4 shows for both different stack sizes and number of positional lags, `RAFFT` execution time against the sequence length. For both stack size and number of positional lags, the execution time scales almost with the same exponent (≈ 2).

In sum, `RAFFT`'s performance shows a significant improvement compared to three folding tools (`RNAfold`, `RNAstructure`, and `ContraFold`), and we can approximate its theorical time complexity to $O(L^2 \log L)$, where L is the sequence length. However, its average CPU time scales with respect to the stack size and the number of positional lags considered. When $N = 1$ and $n = 100$, `RAFFT` CPU time is lower than all of the four tools except for sequences longer than 10^4. But when considering $N = 50$ stacks, `LinearFold` showed better performance. Fitting the empirical CPU times of each tool to a non-linear regression showed that all

the methods scaled with respect to the sequence length whereas, `LinearFold` scales linearly (i.e. $L = 1$) followed by `RAFFT` with an exponent of $L \approx 2$, the MFE prediction methods scale cubically. Now, does the improvement in CPU time guarantee the quality of the predictions? The following section analyses the quality of the structure predictions.

3.2.2 *Accuracy of the predicted structural ensemble*

After comparing `RAFFT`'s computational time to existing tools, it is also essential to assess the quality of the predicted secondary structures. The quality of each tool's predictions is measured using two statistical metrics: the PPV and the sensitivity. This section presents the quality comparison of `RAFFT` predictions to the four previously mentioned tools, i.e. `RNAfold`, `RNAstructure`, `LinearFold`, `Contrafold` and the ML method `Mxfold2`.

We started by analyzing the prediction performances with respect to sequence lengths: we averaged the performances at fixed sequence length. Figure 3.5 shows the performance in PPV and sensitivity for the five methods. It shows that the ML method (`Mxfold2`) consistently outperformed `RAFFT` and the other predictions. When comparing only the MFE predictions produced using the DP tools, `LinearFold` outperformed all other tools (`RNAfold` and `RNAstructure`) for both short and long sequences. The t-test between the ML and the most used MFE prediction tool (`RNAfold`) revealed not only a significant difference (p-value $\approx 10^{-12}$) but also a substantial improvement of 14.5% in PPV. `RAFFT` showed performances similar to `RNAfold`; but, `RAFFT` is significantly less accurate (p-value ≈ 0.0002), with a drastic loss of performance for sequences of length greater than 300 nucleotides (See also Table 3.1).

However, are there relevant structures in the ensemble predicted by our method? To address this question we retained the structure with the best score among the 50 recorded structures per sequence. We obtained an average PPV of 60.0% and an average sensitivity of 62.8% over all the dataset. The gain in terms of PPV/sensitivity is especially pronounced for sequences of length ≤ 200 nucleotides, indicating the presence of biologically more relevant structures in the predicted ensemble than the thermodynamically most stable one (PPV was =79.4%, and sensitivity=81.2%). The average scores are shown in Table 3.1. We also investigated the relation to the number of bases between

paired bases (base pair spanning), but we found no striking effect, as already pointed out in one previous study [1].

Table 3.1: **Average performance displayed in terms of PPV and sensitivity.** The metrics were first averaged at fixed sequence length, limiting the over-representation of shorter sequences. The first two rows show the average performance for all the sequences for each method. The bottom two rows correspond to the performances for the sequences of length ≤ 200 nucleotides.

	RNAfold	LinearFold	RNAstructure	CONTRAfold	Mxfold2	RAFFT	RAFFT*
	All sequences						
PPV	55.9	60.6	54.7	58.4	70.4	47.7	60.0
Sensitivity	63.3	58.9	61.5	65.2	77.1	52.8	62.8
	Sequences with lengths ≤ 200						
PPV	59.5	63.2	58.2	60.5	76.7	57.9	79.4
Sensitivity	65.5	59.4	63.8	65.9	82.9	63.2	81.2

All methods performed poorly on two groups of sequences: one group of 80 nucleotides long RNAs, and the second group of around 200 nucleotides (three examples of such sequences are shown in the Appendix A3.1). Both groups have large unpaired regions, which for the first group lead to structures with average free energies 9.8 kcal/mol according to our dataset. The PCA analysis of the native structure space, shown in Figure 3.6, reveals a propensity for interior loops and the presence of large unpaired regions like hairpins or external loops. Figure 3.6 shows the structure space produced by Mxfold2, which seems close to the native structure space. In contrast, the structure spaces produced by RAFFT and RNAfold are similar and more diverse.

In summary, we performed the prediction quality comparison for different sequence lengths. The dataset was divided into two sets: one with lengths less than 200 nucleotides and the rest constituting the second. Because RAFFT predicts an ensemble of structures, which contrasts the other tools, we also distinguish the single prediction (RAFFT) comparison from the ensemble one (RAFFT*). Overall, on average, RAFFT performed qualitatively poorer than existing tools in terms of both PPV and sensitivity. The ML method, Mxfold2 outperformed all existing methods for different RNA sequence lengths but equalized RAFFT* performance for sequences of length less than 200 nucleotides. The later showed

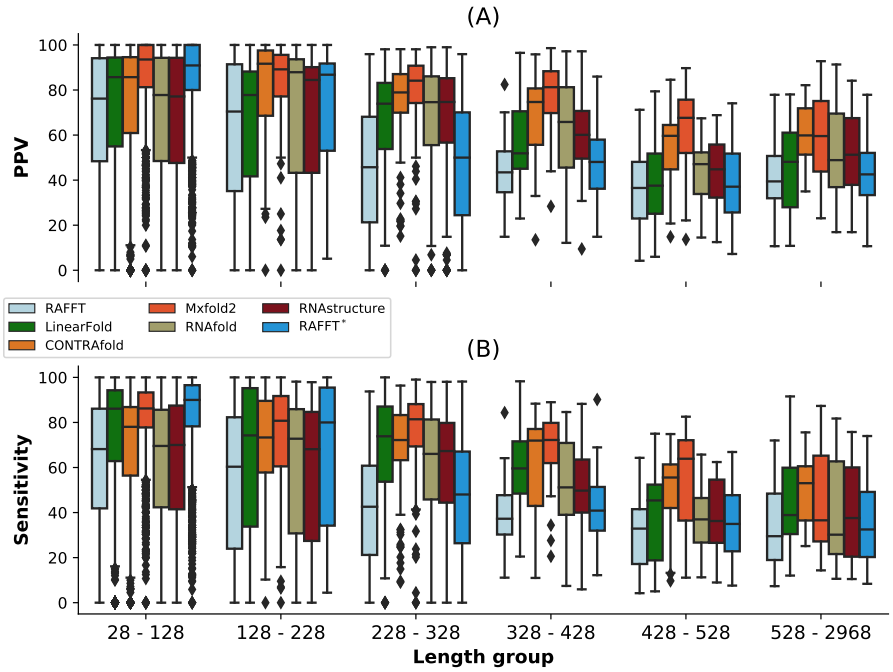


Figure 3.5: **RAFFT's performance on folding task.** (A) PPV *vs* sequence length. In the top panel, RAFFT (in light blue) shows the PPV score distributions when for the structure (out of $N = 50$ predictions) with the lowest free energy, whereas RAFFT* (in blue) shows the best PPV score in that ensemble. (B) Sensitivity *vs* sequence length.

that RAFFT predicted ensemble contains sequences of biological interest. We further assess the quality of that ensemble with the proposed kinetics ansatz. The next section discusses two RNA kinetic test cases: the application of the kinetic ansatz on CFSE and a bistable RNA sequence.

3.2.3 *Applications to the RNA kinetics*

Furthermore, the ensemble of structures predicted by RAFFT is analyzed using a kinetics ansatz to extract information about the dynamic of RNA folding. This section analyses the kinetics of two RNA sequences using RAFFT predicted pathways.

We started with the CFSE, a natural RNA sequence of 82 nucleotides with a structure determined by sequence analysis and obtained from the RFAM database. This structure has a pseudo-knot which is not taken into account here.

Figure 3.7A and Figure 3.7B show respectively the fast-folding graph constructed using RAFFT, and the MFE and native structures

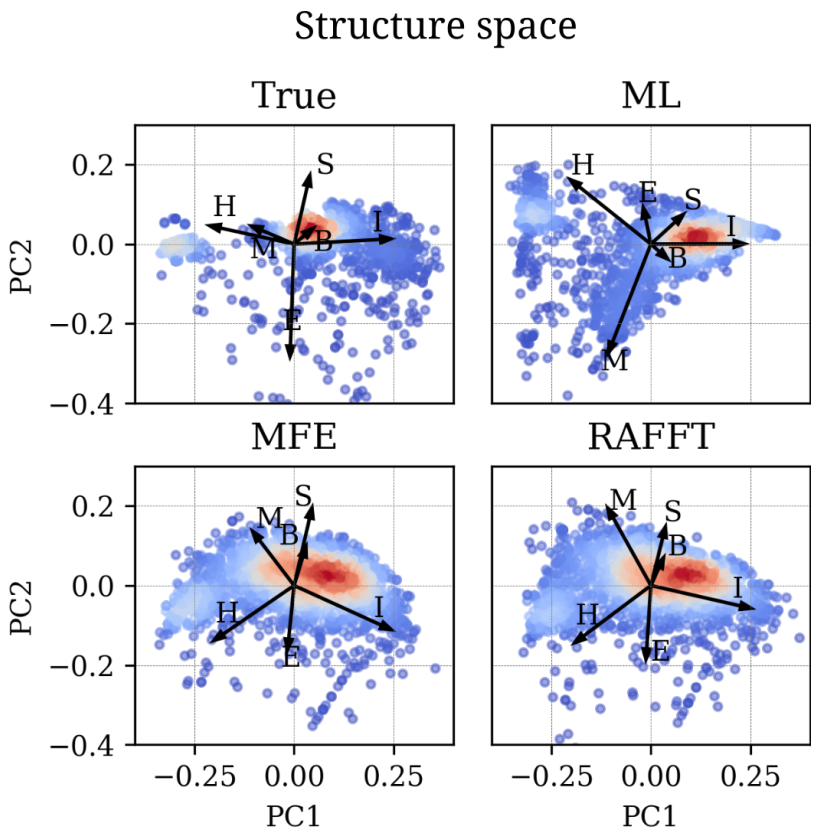


Figure 3.6: **Structure space analysis.** PCA for the predicted structures using `RAFFT, RNAfold, MxFold2` compared to the known structures denoted "True".

for the CFSE. The fast-folding graph is computed in four steps. At each step, stems are constructed by searching for $n = 100$ positional lags and, a set of $N = 20$ structures (selected according to their free energies) are stored in a stack. The resulting fast-folding graph consists of 68 distinct structures, each of which is labelled by a number. Among the structures in the graph, 6 were found similar to the native structure (16/19 base pairs differences). The structure labelled "29" in the graph leading to the MFE structure "59" is the 9^{th} in the second stack. When storing less than 9 structures in the stack at each step, we cannot obtain the MFE structure using `RAFFT`; this is a direct consequence of the greediness of the proposed method. To visualize the energy landscape drawn by `RAFFT`, we arranged the structures in the fast-folding graph onto a surface according to their base-pair distances; for this we used the multidimensional scaling algorithm implemented in the `scipy` package. Figure 3.7D shows the landscape interpolated with all the structures found; this landscape illustrates the bi-stability

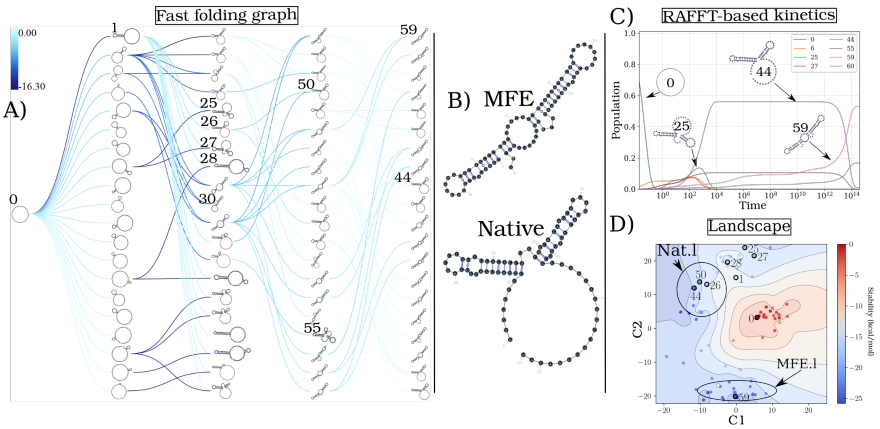


Figure 3.7: **Application of the folding kinetic ansatz on CFSE.** (A) Fast-folding graph in four steps and $N = 20$ structures stored in a stack at each step. The edges are coloured according to $\Delta\Delta G$. At each step, the structures are ordered by their free energy from top to bottom. The minimum free energy structure found is at the top left of the graph. A unique ID annotates visited structures in the kinetics. For example, "59" is the ID of the MFE structure. (B) MFE (computed with `RNAfold`) and the native CFSE structure. (C)The change in structure frequencies over time. The simulation starts with the whole population in the open-chain or unfolded structure (ID 0). The native structure (**Nat.1**) is trapped for a long time before the MFE structure (**MFE.1**) takes over the population. (D) Folding landscape derived from the 68 distinct structures predicted using `RAFFT`. The axes are the components optimized by the MDS algorithm, so the base pair distances are mostly preserved. Observed structures are also annotated using the unique ID. MFE-like structures (**MFE.1**) are at the bottom of the figure, while native-like (**Nat.1**) are at the top.

of the CFSE, where the native and MFE structures are in distinct regions of the structure space.

From the fast-folding graph produced using `RAFFT`, the transition rates from one structure in the graph to another are computed using the formula given in Equation 3.6. Starting from a population of unfolded structure and using the computed transition rates, the native of structures is calculated using Equation 2.3. Figure 3.7C shows the frequency of each structure; as the frequency of the unfolded structure decreases to 0, the frequency of other structures increases. Gradually, the structure labelled "44", which represents the CFSE native structure, takes over the population and gets trapped for a long time, before the MFE structure (labelled "59") eventually becomes dominant. Even though the

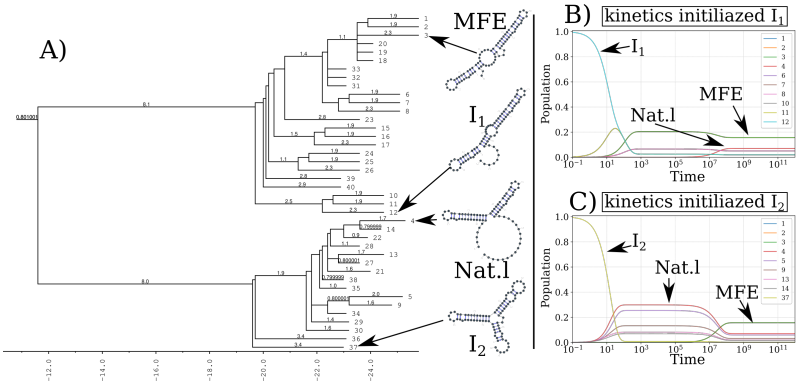


Figure 3.8: **Folding kinetics of CFSE using `Treekin`.** A) Barrier tree of the CFSE. From a set of 1.5×10^6 sub-optimal structures, 40 local minima were found, connected through saddle points. The tree shows two alternative structures separated by a high barrier with the global minimum (MFE structure) on the right side. (B) Folding kinetics with initial population I_1. Starting from an initial population of I_1, as the initial frequency decreases, the others increase, and gradually the MFE structure is the only one populated. (C) Folding kinetics with initial population I_2. When starting with a population of I_2, the native structure (labelled **Nat.1**) is observable, and gets kinetically trapped for a long time due to the high energy barrier separating it from the MFE structure.

fast-folding graph does not allow computing energy landscape properties (saddle, basin, etc.), the kinetics built on it reveals a high barrier separating the two meta-stable structures (MFE and native).

Our kinetic simulation was then compared to `Treekin` [55]. First, we generated $1.5{\times}10^6$ sub-optimal structures up to 15 kcal/mol above the MFE structure using `RNAsubopt` [112]. Since the MFE is $\Delta G_s = -25.8$ kcal/mol, the unfolded structure could not be sampled. Second, the ensemble of structures is coarse-grained into 40 competing basins using the tool `barriers` [55], with the connectivity between basins represented as a barrier tree (see Figure 3.8A). When using `Treekin`, the choice of the initial population is not straightforward. Therefore we resorted to two initial structures I_1 and I_2 (see Figure 3.8B and 3.8C, respectively). In Figure 3.8B, the trajectories show that only the kinetics initialized in the structure I_2 can capture the complete folding dynamics of CFSE, in which the two metastable structures are visible. Thus, in order to produce a folding kinetics in which the native and the MFE structures are visible, the kinetic simulation performed using

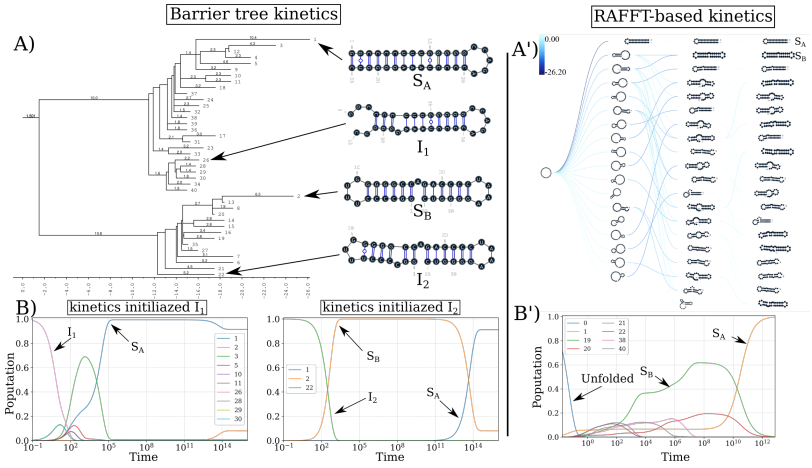


Figure 3.9: **RAFFT *vs* Treekin: folding kinetics of a bi-stable RNA sequence.** (A) Barrier tree for the bi-stable example sequence. The local minima and the corresponding barriers are computed from the complete enumeration of the structure space. The bi-stability is visible on the barrier tree through the two branches separated by a high barrier. (B) Folding kinetics trajectories. The left plot shows the folding dynamics starting from a population with I_1, and the right size is the kinetics when the population is initialized in structure I_2. When starting from I_1, S_A is quickly populated; starting from I_2, the bi-stability is more apparent. (A′) Fast-folding graph using RAFFT. A maximum of $N = 20$ structures are stored in a stack at each step and overall 46 distinct structures are visited. (B′) Folding kinetics trajectory obtained from the fast-folding graph (indices are different from the barrier tree indices). The dynamics starts with a population with only unfolded structure, and slowly, S_B is populated and gets trapped for a long time before the MFE structure S_A becomes populated.

Treekin required a particular initial condition and a barrier tree representation of the energy landscape built from a set of 1.5×10^6 structures. By contrast, using the fast-folding graph produced by RAFFT, which consists only of 68 distinct structures, our kinetic simulation produces complete folding dynamics starting from a population of unfolded structure.

As a second illustrative example, we applied both kinetic models to the classic bi-stable sequence. For Treekin, we first sampled the whole space of 20×10^3 sub-optimal structures from the unfolded state to the MFE structure, and from that set, 40 basins were also computed using barriers. The barrier tree in Figure 3.9 shows the bi-stable landscape, where the two deepest

minima are denoted S_A and S_B. As in the first application, we also chose two initializations with the structures denoted I_1 and I_2 in Figure 3.9A and 3.9B. Secondly, we simulate the kinetics starting from the two initial conditions (See Figure 3.9B). When starting from I_2, the slow-folding dynamics is visible: S_B first gets kinetically trapped, and the MFE structure (S_A) only takes over later on. For our kinetic ansatz, we started by constructing the fast-folding graph using RAFFT, consisting of only 46 distinct structures. The resulting kinetics, shown in Figure 3.9B' was found qualitatively close to the barrier kinetics initialized with structure I_2. Once again, with few as 48 structures, our proposed kinetic ansatz can produce complete folding dynamics starting from a population of unfolded structure.

In both examples, our kinetic ansatz derived from the fast folding graph predicted by RAFFT produces complete folding kinetic trajectories, using fewer structures than the existing methods that required the complete enumeration of the fitness landscape (i.e. all structures and their associated energies). Despite the poor validation procedure of our kinetic ansatz, we believe that the RNA pathways predicted by RAFFT could contain structures of biological pertinence. An analysis of the sample structures produced by RAFFT is provided in Appendix Section A.1 and a discussion on some limitations in Chapter 6.

3.3 CONCLUSION

We have proposed a method for RNA structure, and dynamics predictions called RAFFT. Our method is inspired by the experimental observation of parallel fast-folding pathways. To mimic this observation, we designed an algorithm that produces parallel folding pathways in which stems are formed sequentially. Taking advantage of the FFT, the time complexity of our method was slowed down to $O(L^2 \log L)$, thus improving the cubic time complexity of classic DP methods. Then, we proposed a kinetic ansatz that exploits the parallel fast-folding pathways predicted to model how different conformations are populated over time. Our kinetic ansatz produced complete folding dynamics without sampling the entire conformation space. However, our method also presents some limitations that will be discussed in Chapter 7.

Part II

RNA DESIGN

This second part of our thesis fucuses only on the inverse folding of RNA secondary structures. It contains figures and ideas that have previously appeared in our publications:

- [128] **Nono SC Merleau** and Matteo Smerlak (2021). *A simple evolutionary algorithm guided by local mutations for an efficient RNA design*. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 1027-1034. (Published)
- [129] **Nono SC Merleau** and Matteo Smerlak (2022). *An evolutionary algorithm for inverse RNA folding inspired by Lévy flights* In: *bioRxiv* (Submitted and accepted) (BMC Bioinformatics).

4

INTRODUCTION TO RNA DESIGN

The previous chapters demonstrated the implications of ncRNAs molecules in varying levels of cellular processes, from gene expression regulation (miRNAs, piRNAs, lncRNAs) to RNA maturation (sncRNAs, snoRNAs) and protein synthesis (rRNAs, tRNAs). Knowing that these biological functions are performed by high dimensional RNA structures, which strongly depend on their secondary structures, we also provided a comprehensive review of computation methods for predicting secondary structures. Now that we have computational folding tools that are accurate enough, is it possible to design an RNA molecule that can accomplish a desired biological function for a given secondary structure? Answering this question may demand both experimental and computational efforts. For artificial ncRNAs for which the native RNA sequence is unknown, the essential prerequisite for experimentalists is often a computational solution to the inverse folding problem. Unlike the folding situation, the inverse folding problem begins from a given secondary structure, and the goal is to find one or many RNA sequences that fold into that secondary structure. This chapter aims to provide the formal background and biotechnological implications of addressing this problem. Then, it gives a brief literature review of the existing computational methods.

4.1 RNA INVERSE FOLDING AND BIOTECHNOLOGICAL IMPLICATIONS

In modern biotechnology, we often seek to reproduce the natural ability of the cells to control gene expressions using a variety of nucleic acids and proteins. These natural cellular abilities result from networks of regulatory molecules such as ncRNAs that dynamically regulate the expression of specific genes in response to environmental signals. Therefore, the ability to engineer biological systems is directly related to controlling gene expression. The increasing number of examples of natural regulator ncRNAs has opened doors to many emerging subfields such as RNA synthetic biology [21, 86] and RNA nanostructure [68, 89]. Researchers have engineered RNA molecules with new biological functions, in-

spired by this natural versatility. Synthetic biology has also made significant progress in developing versatile and programmable genetic regulators that precisely control gene expressions in the last decades. Three general approaches are taken to engineer new functional RNAs: harvesting from nature, computational design and molecular evolution. We are interested here in computational RNA design methods.

In most cases, designing a functional RNA goes beyond computationally generating a set of RNA sequences that fold into a given secondary structure. Successful design methods include computational and experimental, predictive and analytical techniques. However, computational tools addressing the inverse folding problem often provide some guidance and rationalities through the design process. For example, Steffen Mueller and his collaborators [134] suggested a systematic, rational approach, synthetic attenuated virus engineering (SAVE), to develop new, productive live attenuated influenza virus vaccine candidates using computer-aided rational design. In addition, Eckart Bindewald et al. [13] used computational tools for solving inverse RNA folding in the design of nanostructures, including pseudoknots. And in designing several ncRNAs with a successful synthetic such as ribozymes [40], riboswitches [52, 203].

Depending on the specificities of the RNA design task, finding the underlying mathematical model that maps each designed RNA sequence solution to a set of properties that includes most of the specifications or constraints can be a challenging task. When it exists, it allows to address the RNA design problem computationally, and we call this mathematical model the objective function of the RNA design problem. The complexity of the objective function used gives rise to two RNA design problems: the negative and the positive design. The following section describes both RNA design problems and their computational complexities.

4.2 THE POSITIVE AND NEGATIVE DESIGN

We often find two types of RNA design problems in the literature: negative and positive design. The negative structural design of RNAs, also called the inverse RNA folding problem, aims to find one or many RNA sequences that fold into a given target RNA secondary structure while avoiding alternative folds of similar quality for the chosen energy model ΔG. In other terms, it is an optimization problem where a target RNA secondary structure $\mathcal{S}^*$

of length L is given, and the goal is to determine an RNA sequence ϕ of length L such that $\forall \mathcal{S} \neq \mathcal{S}^* \in \Sigma_\phi, \Delta G(\phi, \mathcal{S}) > \Delta G(\phi, \mathcal{S}^*)$.

This problem is NP-hard even in a simple energy model [14], and we cannot provide a parameterized algorithm that solves it in a polynomial time.

In contrast, a positive design problem consists of optimizing affinity towards a given target secondary structure. In other terms, the objective is to find a sequence $\phi \in \{A, U, C, G\}^L$ such that $\mathcal{S}^* = \mathcal{S}^{MFE}(\phi) = \arg\min_{\mathcal{S} \in \Sigma_\phi} \Delta G(\phi, \mathcal{S})$ (i.e. the sequence ϕ should have as MFEs structure of its ensemble Σ_ϕ the target structure $\mathcal{S}^*$). The positive design is computationally solvable exactly in polynomial time [54].

Both negative and positive designs are considered in this work, and the main difference often depends on the objective function used. In addition, it has been recently shown that the proportion of designable secondary structures decreases exponentially with L for various popular combinations of energy models and design objectives [216]. The following section presents an overview of previously used objective functions of for the RNA design problem.

4.3 OBJECTIVE FUNCTIONS PREVIOUSLY USED IN THE CONTEXT OF INVERSE RNA FOLDING

For a given target secondary structure $\mathcal{S}^*$ of length L, a brute force approach to the inverse RNA folding problem that enumerates all possible RNA sequences is not viable due to the exponential growth of the search space with increasing length (i.e. 4^L). For the space of compatibles sequences to the target $\mathcal{S}^*$, an upper bound can be defined by restricting the paired position to the base pairs: G-C, G-U, and A-U. This results in $6^{(L-u)/2} \times 4^u$ sequences compatible with $\mathcal{S}^*$ where u is the number of unpaired nucleotides. The most common way to efficiently handle the huge set of possible solutions is to solve an optimization problem subjected to a formulated objective function. There exists a variety of well-established optimization methods helping to perform this task. However, finding the right objective function to evaluate the solutions can be quite challenging. This section of our work provides an overview of an objective function and an essential description of the most previously used objective functions in designing RNA molecules.

The objective function defines a mathematical model that maps each RNA sequence solution to its essential properties or func-

tions. In biological terms, this relation between fitness and sequence can be seen as assigning a phenotype (score) to a genotype (sequence). Selection pressure due to the optimization method ensures that better phenotypes are advantageous and thus preferred, which optimizes the sequence to fall into fitness optima. This section defines the previously used objective functions in the RNA design problems and highlights some interesting properties.

- A simple distance from the target structure: in the simplest setting, the objective function of an RNA sequence ϕ defines the distance between $\mathcal{S}^*$ and the current MFE structure $\mathcal{S}^{MFE}(\phi)$. It often requires only the MFE structure's computation, hence being computationally fast. There are many variants of this distance measure: base-pair distance, hamming or string edit distance, tree-edit distance and energy distance. For a formal definition of each of those distances, see Section 1.4. This objective function was used in the earliest tools such as `RNAinverse` [80] but also in many others since then [5, 18, 59] .

- A negative design objective function: in contrast to the above mentioned objective functions (often considered when performing a positive design), we consider the whole structural ensemble when computing the fitness of an RNA sequence ϕ. In most cases, it is preferable also to consider negative design goals, which allows for avoiding alternative structures of similar quality to the target structure. Negative RNA design methods usually consider one of the three following defects: (1) the *suboptimal defect* [37, 54, 80, 218] which defines the energy distance to the first suboptimal (2) the *probability defect* [80, 218] which defines the probability that the sequence ϕ folds into any other structure than the target structure $\mathcal{S}^*$ and (3) the *ensemble defect* [218] which corresponds to the average number of incorrectly paired nucleotides at equilibrium calculated over the structure ensemble of ϕ, Σ_ϕ.

- Multi-objective optimization: in some designing cases where more than one goal is specified, it is necessary to formulate an objective function for each goal. That results in a multi-objective optimization problem. The solutions to such a problem are all optimal for at least one objective function and thus arranged on the so-called Pareto optimal front. This approach has already been used in several RNA design tools such as `Modena` [189, 190] and in [143].

- Bistable and multi-stable riboswitches objective functions: In some designing cases, especially for riboswitches, it is possible to specify more than one desired target structure, including the energy differences between them, the barrier heights and the kinetic properties. Following the same idea, Flamm et al. introduced an objective function that enables designing RNA molecules to adopt two distinct structures [54]. This bistable objective function contains two terms. The first term increases the probability of both structures in the ensemble, and the second specifies the desired energy difference between both states. It is also possible to vary the states' temperature to gain a bistable thermoswitch. The same idea has therefore been expanded to an objective function for designing RNA molecules that can adopt more than two structures, including extension for multi-structure energy barrier calculations [143, 173]. `Frnakenstein` [117] also utilises such objective function for multi-target design.

- Mutational robustness and neutrality: In addition to the above-mentioned objective functions, objective functions aim to measure the mutual neutrality of the sequence concerning the target structure [173]. When using such an objective function, the sequences are optimized so that the fraction of one-mutant neighbours to the original structure is as significant as possible. This allows for perfectly preserving the structure when mutations are introduced. We often talk of a mutational robustness optimization [6].

These objective functions suggest that the inverse folding problem is a major challenge with no single solution yet, and many possible ways of setting the goal. This thesis relies on three objective functions: the simple distance to the target, the ensemble defect, and the mutational robustness. In addition to the many objective functions, there are also several methods. The following section will review the existing methods independently of the objective function and provide some limitations.

4.4 A REVIEW ON EXISTING INVERSE RNA FOLDING TOOLS.

Several methods or algorithms addressing this problem have been proposed in the literature. The existing techniques can be classified into two categories: one for the pseudoknot-free structure design and another for the pseudoknotted RNA structure

design. This section gives a short description of some of the existing tools, especially those used in the benchmark results of the thesis.

4.4.1 *Pseudoknot-free RNA inverse folding tools*

Due to the complexity of the RNA design, most of the existing tools perform a stochastic search optimization where initial potential solutions are generated and refined over a finite number of iterations or generations [43, 47, 48, 141, 188]. Some stochastic search techniques may involve several candidate solutions at each generation or not. The ones that do are population-based algorithms, which means they maintain a set of candidate solutions at each generation, with each solution corresponding to a unique point in the problem's search space. We are interested in this work in EA, a particular class of population-based algorithms. This section presents an overview of EA when applied to the inverse folding of RNA molecules. In addition, it reviews the existing tools implementing similar and different techniques.

4.4.1.1 *Evolutionary algorithms and RNA inverse problems*

Among the existing tools dealing with the RNA inverse problem, both `ERD` [47, 48] and `MODENA` [189] are EAs but implementing different strategies. In general, an evolutionary search algorithm on any fitness landscape consists of three main parts, which in the context of RNA inverse folding are as follows:

- Initialization: generating a random initial population of RNA sequences compatible with the given target secondary structure.
- Evaluation and selection: evaluating a population of RNA sequences consists of two steps: 1) fold each sequence into a secondary structure and assign it a weight based on its similarity to the target structure. 2) select a weighted random sample with replacement from the current population to generate a new population. A detailed description of the objective function used in our proposed tool `aRNAque` is provided in the next chapter.
- Mutation (or move) operation: define a set of rules or steps used to produce new sequences from the selected or initial ones. This component is elaborated further in the next chapter.

MODENA uses a multi-objective function that measures the stability of the folded sequence and its similarities to the target. It starts from a population of randomly generated sequences, and the objective is optimized through tournament selection and random mutation at non-closing loop positions.

In contrast, ERD starts by decomposing the target structure into loops and independently uses an evolutionary algorithm to minimize each constituent's energy. It was first developed in 2014 [48], and one year after, an updated version was released [47]. The main lines of ERD are:

1. Pool reconstruction: using a collection of RNA sequences (STRAN database) similar to the natural ones, a pool of sequences is constructed for their length by successively finding the corresponding structure using RNAfold, decomposing the structure in sub-components, and finally, the corresponding sub-sequences of the same size are gathered to form a pool.

2. Hierarchical decomposition of the target structure into loops: using the idea that any secondary structure can be uniquely decomposed into its structural components (stems, hairpin loops, internal loops, bulge and multi-loops), ERD decomposes the target in the positions where multi-loops occur.

3. Sequence initialization: after decomposing the target structure into sub-components, for each sub-component, a random sub-sequence is chosen from the pool, and the initial sequence is a combination of those sub-sequences;

4. Evolutionary optimization of the sub-sequences: an EA algorithm is performed on each sub-component to improve the initial sequence. The outcome sub-sequences are combined to form a newer sequence that will replace the initial one. Iteratively the evolutionary algorithm is performed on the updated sequence until the combined sequence folds into the target or in a failure case when the stopping condition is satisfied. Two evolutionary operators are implemented here, a mutation that consists of replacing a subsequence corresponding to a sub-component with a new random one from the pool for the same length, and a selection which consists of choosing from a population of 15 RNA sequences or sub-sequences, three best sequences with respect to their free energy and adding them to the best from the preview generation, three best ones with respect

to the Hamming distance from the target are therefore chosen. The next-generation population is then obtained by generating five new sequences for each of the three best sequences.

In the different EA methods presented above, the mutation operation is essential for good performance because it provides the rules that allow for navigating the solution space. `ERD` implements a non-local mutation, which consists of randomly changing a subsequence in the candidate solution with a new one taken from a set of possible moves. In contrast, `Modena` uses both local mutation and crossover operation to improve its search. However, both EAs present difficulties in finding RNA sequences that fold into some secondary structures of the `Eterna100` data set. That limitation could be due to the local search (for `Modena`) or the finite set of move data used in the non-local search implemented in `ERD`. In mathematical optimization, local searches are known for their quick convergence to a local minimum. This could be the same case for EAs implementing local mutations. To avoid early convergence EA practitioners often implement non-local mutation methods, e.g. Lévy search, inspired by the Lévy flights. The following section describes the Lévy flight and reviews some applications of Levy search in the context of EAs.

4.4.1.2 *Lévy flights and evolutionary algorithms*

In this section, we define concepts such as Lévy flights and provide a brief review of its implications and applications to optimization techniques such as evolutionary algorithms.

In its classical setting, evolutionary algorithms are guided by local (or one-point mutations) mutations. Although a local search can efficiently discover optima in a simple landscape, more complex landscapes pose challenges to designing evolutionary algorithms that rely solely on local search. This is especially true on a landscape with high neutrality where local search may be inefficient or risk getting stuck on a plateau (or local optimum). To avoid this pitfall, many practitioners suggested EA that implements a mutation scheme inspired by Lévy flights (called Lévy mutation).

Lévy flights are random walks with a Lévy (or any heavy-tailed) step size distribution. The concept originates in the work of Mandelbrot on the fluctuation of commodities prices in the 1960s [120] but has since found many more physical applications [172]. The term "Lévy flight" was also coined by Mandelbrot, who

used one specific distribution of step sizes (the Lévy distribution, named after the French mathematician Paul Lévy). Lévy flights also play a key role in animal foraging, perhaps because they provide an optimal balance between exploration and exploitation [93, 201]. For a recent review of applications of Lévy flights in biology from the molecular to the ecological scale, [149].

Similar to a Lévy flight, a Lévy mutation scheme allows simultaneous search at all scales over the landscape. New mutations most often produce nearby sequences (one-point mutations), but occasionally generate mutant sequences which are far away in genotype space (macro-mutations). In this work, the distribution of the number of point mutations at every step is taken to follow a Zipf distribution [135].

Earlier works have applied similar ideas in genetic programming [30], and in differential evolutionary algorithms [170]. This motivated us to investigate a possible benefit of a Lévy flight in the design of RNA sequences in Chapter 5. In addition to EA methods, there exists several computational RNA design tools implementing different techniques such as, ML, nested monte carlo search (NMCS) etc... The following section provides a short description of such tools.

4.4.1.3 *Tools implementing non-EA strategies.*

Several tools dealing with the RNA folding problem implement different strategies from the population-based, or evolutionary algorithm approaches. This section describes couple of them, emphasising on those that are used in the benchmark results in Chapter 5, which are `NEMO`, `RNAinverse`, `antaRNA` and `sentRNA`.

`sentRNA` [171] is a computational agent that uses a set of information and strategies collected from the `EteRNA` game players to train a neural network model. The neural network assigns an identity of A, U, C, or G to each position in the given target, a featured representation of its local environment. The featured representation combines information about its bonding partner, nearest neighbours, and long-range features. While the bonding partner and nearest neighbour information are provided to the agent by default, long-range features are learned through the training data. For each target structure, the long-range features refer to the important position j relative to i that the agent should know about when deciding what nucleotide to assign to i. These are defined by two values: the Cartesian distance and the angle in radians. Those two values are computed for each position (i, j) us-

ing a mutual information metric over the player solution dataset. Therefore, the result is a list of long-range features for a given target structure. A subset of long features is selected from this list and used to define a model for the neural network model's training, validation, and testing. In addition to the neural network model, `sentRNA` also implements a refinement algorithm on the unsuccessful design. The refinement algorithm is an adaptive walk that starts from the predicted sequence and uses a set of random mutations that allow improving the neural network solution. Alternatively, `EternaBrain` [105] implement a convolutional network model trained on a huge `EteRNA` moves-select repository of 30,477 moves from the top 72 players; and `LeaRNA` [155] uses deep reinforcement learning to train a policy network to sequentially design an entire RNA sequence given a specified target structure.

`NEMO` [141] is a recently developed tool combining a NMCS technique with domain-specific knowledge to create a novel algorithm. The underlying idea is to start with an input pattern sequence of N's of the same length as the targeted structure. First, it uses the standard NMCSs to sample sequence solutions acting on N's only. A sequence candidate is selected from the sample; then folded into an MFE structure. When the MFE structure does not match the target, some subset mutations are performed, and a set of random mutated positions are picked to generate a new input pattern sequence. The new input pattern will allow sampling acting on N's only using the same standard NMCSs. This procedure is then repeated several times until the MFE structure matches the targeted structure or not in the unsuccessful cases. The statistical results show that `NEMO` surpasses all the existing tools on the `EteRNA100` benchmark datasets by solving $\approx 95\%$ of the targets using the Turner1999 energy parameter sets. Using a similar technique, `RNAinverse`[113], one of the oldest inverse folding tools included in the `ViennaRNA` package, uses an adaptive random walk to minimize base-pair distance. The distance is computed by comparing the MFE structure of the mutated sequence with the target structure. In addition, `RNAinverse` allows for designing more probable sequences using the partition function optimization. The latter allows for more stable designed sequences that mostly fold into MFE structures different from the target structure. On an attempt to improve `RNAinverse`, many other tools have been suggested `INFO-RNA` [18], `RNA-SSD` [5] and `DSS-Opt` [126]. The most recent tools also include `RNAPOND` [217] and `MaiRNAiFold` [130].

antaRNA [100] is also a recent program available since 2015, and it provides a web server for friendly usability. It utilizes an *ant-colony* optimization, in which an initial sequence is generated via a weighted random search, and the *fitness* of that sequence is then used to refine the weights and improve subsequences over generations. It provides many other interesting features, such as the sequence and target GC-content constraints. It also provides a fast python script that includes the options from the web server presented through a command line. Other tools also provide this dual advantage but implement different optimization techniques. NUPACK:design [219] uses a tree decomposition technique and the ensemble defect as objective function to design qualitatively good sequences. incaRNAfbinv [44] is a program for fragment-based RNA design. incaRNAfbinv's web server combines two complementary methodologies: IncaRNAtion [146] and RNAfbinv [209]. IncaRNAtion generates a GC-weighted partition function for the target structure, and then adaptively samples sequences from it to match the desired GC-content. RNAiFold [60] employs constraint programming that exhaustively searches over all possible sequences compatible with a given target. RNAiFold [60] has the particularity of designing synthetic functional RNA molecules.

So far, except for Modena and antaRNA, most of the computation tools presented in previous sections do not account for pseudoknotted RNA target structures, which represents a disadvantage, knowing their implications in realizing ncRNA biological functions. The following section reviews existing RNA design tools that support pseudoknotted secondary structures.

4.4.2 *Pseudoknotted RNA inverse folding tools*

Designing RNA sequences for pseudoknotted targets is computationally more expensive than pseudoknot-free targets. For that reason, many of the studies addressing the inverse folding of RNA considered only pseudoknot-free secondary structures. There are, however, some exceptions: MCTS-RNA [215], antaRNA[100], Modena and Inv[59]. The computation tool presented in Chapter 5 of our work also considers pseudoknots. This section gives an overview of each of these tools.

Inv was one of the first inverse folding tools handling pseudoknotted RNA target structures, but it was restricted to a specific type of pseudoknot pattern called 3-crossing nonplanar pseudoknots.

More recently, MCTS-RNA's authors suggested a new technique that deals with a broader type of pseudoknots. It uses a monte carlo tree search (MCTS) technique which has recently shown exceptional performance in Computer Go. The MCTS allows initialising a set of RNA sequence solutions in MCTS-RNA and the solutions are further improved through local updates at the nucleotide positions.

Another approaches (Modena, antaRNA) implements different strategies one which is a multi-objective ant-colony optimisation and the another one which is a multi-objective evolutionary algorithm. Although the first versions were implemented for pseudoknot-free structure [100, 188], they have since been extended to support pseudoknotted RNAs [99, 189].

Each of the tools mentioned above rely on a folding tools that predicts pseudoknotted secondary structure: MCTS-RNA uses pkiss whereas the other tools (antaRNA and Modena) support two folding tools such as HotKnots and IPKnot. In the context of this work, two folding tools are used HotKnots and IPKnot, and they support the two main types of pseudoknot patterns (i.e. H-type and K-type) contained in the benchmark data used to evaluate our result in Chapter 5. Both pseudoknotted and pseudoknot-free benchmark data sets are considered in this work. The following section describes the benchmark data used to evaluate our proposed EA tool.

4.5 BENCHMARKING THE INVERSE FOLDING TOOLS

The validation of the designed RNA sequences using computational methods often requires biological experiments. Because of the high cost of experimental techniques, most investigators limit their guarantee to using benchmark datasets [23] in general. For pseudoknot-free design tools, two benchmark datasets are mostly used in the literature—(*i*) RFAM [1]: a collection of RNA families, each represented by multiple sequence alignments, consensus secondary structures and covariance models—(*ii*) Eterna100 [3]: a collection of hundred RNA secondary structures extracted from the EteRNA Puzzle game[2]. For RNA inverse tools that support pseudoknots, the PseudoBase++[191] dataset is often considered. This section provides references, descriptions and the cleanup procedure applied for the three data sets mentioned above.

1 The Rfam database https://rfam.xfam.org/
2 The EteRNA game https://eternagame.org/

The `Eterna100` dataset [106] is available in two versions and both contain a set of 100 target structures extracted from the `EteRNA` puzzle game and classified by their degree of difficulty. The `Eterna100-V1` was initially designed using `ViennaRNA` 1.8.5, which relies on Turner1999 energy parameters [197]. Out of the 100 target secondary structures, 19 turned out to be unsolvable using the version of `ViennaRNA` 2.4.14 (which relays on the Turner2004 [123]). Subsequently, an `Eterna100-V2` [106] was released in which the 19 targets were slightly modified to be solvable using `ViennaRNA` 2.4.14 and any version that supports the Turner2004 energy parameters. The main difference between the two dataset relay on the energy parameters used to generate the data.

The `non-EteRNA` (a subset of the `RFAM`) dataset in a set of 63 experimentally synthesized targets that Garcia-Martin et al. [60] recently used to benchmark a set of ten inverse folding algorithms, which from our knowledge, is the most recent and comprehensive benchmark of current state-of-the-art methods. The dataset is collected from 3 sources: the first dataset called **dataset A** which contains 29 targets collected from RFAM and also used in [47, 188] and the second called **dataset B** is a collection of 24 targets used in [47] and added to that the 10 structures used in [171].

The `PseudoBase++` is a set of 266 pseudoknotted RNA structures used to benchmark `Modena`. It was initially 342 RNA secondary structures, but because of the redundancy and the non-canonical base pairs 76 structures were excluded. To group the dataset with respect to the pseudoknot motifs, we used the test data from `antaRNA`'s paper. The test data contains 249 grouped into four categories: 209 hairpin pseudoknots (H), 29 bulge pseudoknots (B), 8 complex hairpin pseudoknots (cH) and 3 kissing hairpin pseudoknots (K). Out of the 266 structures, only 185 (with 150 H-type, 3 K-type, 25 B-type and 7 cH-type) structures were included in the test data. So for that reason, we have used only 185 target structures for the pseudoknot motif performance comparison and the 266 structures for the different target lengths performance comparison.

When the benchmark datasets rely on a particular energy parameter set, the performance of a given inverse RNA folding tool evaluated on these datasets will also be related to the choice of the RNA folding tool's energy parameter set. If the benchmark datasets do not rely on a particular energy parameter set, the robustness of the inverse RNA tool will be its capability to perform well on different energy parameter sets.

4.6 CONCLUSION

In summary, the RNA inverse folding problem is still computationally challenging because there are many objective functions and different ways of evaluating computational tools. Solving this problem is particularly interesting in RNA synthetics, RNA nanostructure design, and emerging fields such as bioengineering. We presented a comprehensive literature review of existing computational methods that addressed this problem in this chapter. The existing approaches have some advantages and disadvantages, depending on the techniques implemented. `NUPACK` for example—despite its well-defined objective function—still has difficulty designing sequences for large targets and most of the `EteRNA100` targets. In contrast, `ERD` because of its powerful decomposition method, which allows dealing quickly with large targets (On RFAM 1.0 with target's length between $400 - 1400$) but is still a big challenge to solve more than 65% of the `EteRNA100-V2` using the Turner2004 energy parameter sets. On another side, `NEMO`, one of the most recent tools, can solve more than 90% of the `EteRNA100-V1` dataset using an old version of `ViennaRNA` package, which is based on Turner1999 energy parameter sets [197]. The `sentRNA`'s machine learning model also relied on the same old version of `ViennaRNA` package and, by adding a refinement on the machine learning model, `sentRNA` solves 78% of `EteRNA100`. Without this refinement, `sentRNA` can only solve 48% of `EteRNA100`'s targets, which can represents another limitation. For the EAs `ERD` and `MODENA`, none of them can solve more than 65% of `EteRNA100` using the Turner2004 energy parameter sets.

In the next chapter, we will introduce a simple evolutionary algorithm called `aRNAque` that implements a Lévy mutation and allows significant improvements to the existing tools.

5 AN EVOLUTIONARY ALGORITHM FOR INVERSE FOLDING INSPIRED BY LÉVY FLIGHTS.

In the previous chapter of our work, we presented the RNA design as an optimization problem and provided a significant literature review on the existing tools addressing that problem. We highlighted some limitations of the existing tools, particularly those implementing evolutionary algorithms. One of the main challenges of evolutionary algorithms is to avoid deception, which is the fast convergence to a local optimum. Most EAs' early convergence to a local optimum is due to the local search implementation, which is the consequence of the local mutation scheme.

To avoid this pitfall, an alternative mutation scheme to the classical local search is the Lévy mutation. We propose an evolutionary algorithm that implements a similar Lévy mutation in this chapter but adjusts to the RNA design problem. This mutation scheme is focused on local search but also searches at all other scales to avoid becoming trapped. its long-range search permits designing RNA sequences of higher positional entropy. Our implementation, called `aRNAque` is available on GitHub as a python script. Compared to existing inverse folding tools, the benchmark results show improved performance on both pseudoknot-free and pseudoknotted datasets. Much of materials in this chapter has been previously published in [128, 129].

5.1 MATERIAL AND METHODS

This section provides a detailed description of `aRNAque` algorithm in general and in particular the Lévy mutation scheme implemented.

5.1.1 *`aRNAque`'s mutation operator*

The previous chapters provided an overview of EA, emphasizing its application to the RNA inverse folding problem. One of the essential components of EAs is the mutation operator. Our tool, `aRNAque`, implements a simple EA that uses a Lévy muta-

tion to explore at different scales the solution space. In addition, our mutation allows explicitly controlling the GC-content of the designed RNA sequences. This section presents in detail our proposed mutation operator.

For a given target RNA secondary structure in its string representation σ^* of length L, the space of potential solutions to the inverse folding problem is $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}^L$. An evolutionary algorithm explores the space of solutions through its move (or mutation) operator. To explore the search space of compatible sequences (sequences with canonical base pairs at the corresponding open and closed bracket positions) with σ^* exclusively, we propose a mutation step that depends on the nucleotide canonical base pair probability distribution.

Let $\mathcal{N} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$ be the set of nucleotides weighted respectively by the probabilities:

$$P_{\mathcal{N}} = \{w_{\mathrm{A}}, w_{\mathrm{C}}, w_{\mathrm{U}}, w_{\mathrm{G}}\}$$

and $\mathcal{C} = \{\mathrm{AU}, \mathrm{UA}, \mathrm{CG}, \mathrm{GC}, \mathrm{UG}, \mathrm{GU}\}$ be the set of canonical base pairs weighted respectively by the probabilities:

$$P_{\mathcal{C}} = \{w_{\mathrm{AU}}, w_{\mathrm{UA}}, w_{\mathrm{CG}}, w_{\mathrm{GC}}, w_{\mathrm{UG}}, w_{\mathrm{GU}}\}$$

where

$$\sum P_{\mathcal{N}} = 1, \sum P_{\mathcal{C}} = 1$$

Our evolutionary algorithm relies on the flexibility of the mutation parameters $P_{\mathcal{N}}, P_{\mathcal{C}}$. These parameters allow explicit control of the GC-content of the RNA sequences during the designing procedure.

We examined the binomial and Zipf distributions:

- Binomial mutation: here U has a binomial distribution:

$$P(U = n) = \binom{L}{n} \mu^n (1-\mu)^{L-n}$$

  for some $0 \leq \mu \leq 1$, such that $u = \mu \cdot L$. We can think of this mutation mode arising from each nucleotide of an RNA sequence independently undergoing a point mutation with probability μ, i.e. μ is the per-nucleotide or point mutation rate.

- Lévy mutation: U has a Zipf distribution given by:

$$P(U = n) = \frac{1/n^c}{\sum_{k=1}^{L} 1/k^c}$$

where $c > 0$ is the value of the exponent characterizing the distribution.

Figure 5.1 shows the distribution of the number of point mutations on a sequence of length 88 nucleotides for both mutation schemes. Both distributions have the same mean, and the difference between the two distributions is more perceptible on their tails.

In the rest of this work, a local mutation will refer to a binomial mutation with parameter $\mu \approx 1/L$.

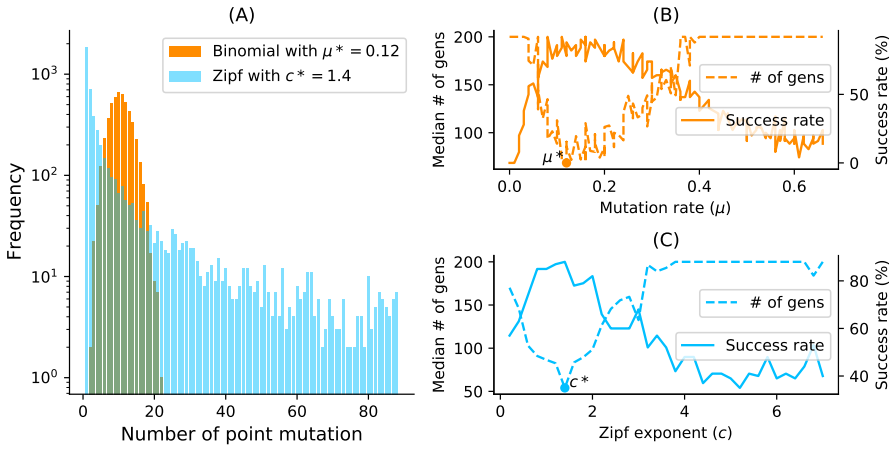


Figure 5.1: **Binomial *vs.* Zipf distributions.** (A) Samplings Binomial and Zipf distributions for the best binomial mutation rate μ^* (respectively c^* for the best Zipf exponent parameter). Both distributions have a mean of 8.7 point mutations for a sequence of length 88 nucleotides. (B) Tuning of binomial mutation rate parameter. For each $\mu \in [0, 1]$ with a step size of 0.005 and the pseudoknotted target `PKB00342` of length 88, 50 sequences were designed using `aRNAque`. (B) shows the median generations and the success percentage *vs.* the mutation rate (μ). The best mutation rate is $\mu^* = 0.085$ (with a median number of generation 93.5 and a success rate of 92%). (C) Tuning of Levy exponent. Similar to (B), for each $c \in [0, 7]$ with a step size of 0.1 and for the same pseudoknotted target structure, 100 sequences were designed using `aRNAque`. It shows the median generations and the percentage of success *vs.* the exponent parameter (c). The Zipf exponent distribution that produced the highest success rate and the minimum number of generations is $c^* = 1.4$.

.

We present the mutation algorithm in algorithm 1. This mutation algorithm is intergraded in a unified EA framework, allowing

to update RNA sequence solution at each iteration or generation. After we apply the mutation operation to the population of RNA sequences, we evaluate the newly generated population; this is usually done using an objective or fitness function. In the following section, we describe the different objective functions taken into account in our implemented EA.

5.1.2 *aRNAque's objection functions*

Our EA reaches its performance through the minimization of three objective functions:

- Hamming distance from the target structure: Since the main goal of the inverse folding problem is to find sequences that fold into a given target secondary structure σ^*, the simple fitness measurement f of an RNA sequence ϕ can be defined as follows:

$$f(\phi, \sigma^*) = \frac{1}{1 + d_h(\sigma^{MFE}(\phi), \sigma^*)} \tag{5.1}$$

  where $d_h(\cdot, \cdot)$ is the hamming distance on the structure space (structures are in dot and bracket representation) defined in Equation 1.12.

- normalyzed energy distance (NED): It is used to minimize the free energy of the designed sequences. (See Equation 1.14)

- ensemble defect (ED) [219]: Here, we use the ED as a second objective function for refinement after having at least one sequence that folds into the target in the current population. It is defined in Equation 1.13.

To minimize the NED and the hamming distance of a population of RNA sequences, instead of combining both measurements to form a multi-objective function, we use them separately at a different level of our EA. We use the NED as a selection weight for the sequences that will be mutated, and the hamming distance is used as a weight to elite ten best sequences that will always move to the next generation. Therefore the selection method we use is the *fitness proportionate selection*, also known as roulette wheel selection [110]. Once we have at least one sequence that folds into the given target in the current population (for the successful case), we start a random walk in its neutral network by minimizing

Algorithm 1: aRNAque's mutation algorithm

/* $P' = \{\phi'_1 \dots \phi'_n\}$: the mutated population;
$P = \{\phi_1 \dots \phi_n\}$: a list of n RNA sequences to mutate;
$P_C = \{w_{AU}, w_{UA}, w_{CG}, w_{GC}, w_{UG}, w_{GU}\}$: a vector containing the weights associated with each canonical base pairs;
$P_N = \{w_A, w_U, w_C, w_G\}$: a vector containing the weights associated with each nucleotide;
$\mathcal{D}$: a given probability distribution (Lévy or Binomial) with parameter p and L. Where L is the length of the target RNA structure */

Input: P, $\mathcal{D}(p, L)$, P_C, P_N
Output: P'

$\{B_i\} \sim \mathcal{D}(p, L)$, where $i \in \{1, 2, \dots, n\}$; // Draw n random numbers that follows a given distribution $\mathcal{D}(p, L)$ (Lévy or Binomial). B_i is the number base pairs to mutate
$\{U_i\} \sim \mathcal{D}(p, L)$, where $i \in \{1, 2, \dots, n\}$; // Draw n random numbers that follows the same distribution as B_i (Lévy or Binomial). U_i is the number non base pair positions to mutate
for $i \in \{1, 2, \dots, n\}$ **do**
    $\phi' \leftarrow P_i$; // Assign the sequence $\phi_i \in P$ to ϕ'
    for $j \in \{1, 2, \dots U_i\}$ **do**
        $r \in \{1, 2, \dots, L\} \sim \mathcal{U}$; // select uniformly a random position in the RNA sequence ϕ'
        $n_j \in \{A, U, C, G\} \sim P_N$; // select a random nucleotide n_j with respect to P_N
        $\phi'_r \leftarrow n_j$; // replace the nucleotide at position j in the RNA sequence ϕ' with n_j
    for $j \in \{1, 2, \dots B_i\}$ **do**
        $k_j \in \{AU, UA, CG, GC, UG, GU\} \sim P_C$; // select a random base pair k_i with respect to P_C
        $b \in \{(b_1, b_2)_i\} \sim \mathcal{U}$; // select uniformly a random pair of base pair positions
        $\phi'_b \leftarrow k_j$; // replace respectively the nucleotides at the base pair position $b_i \in b$ by $k \in k_j$
    $P' \leftarrow P' \cup \phi'$; // Add ϕ' to the list P'

the ensemble defect function (Equation 1.13). The next section provides more detailed information about the core of our EA and the full pseudo-code.

Now that we have defined the mutation and selection operators implemented in our EA, we will describe the general algorithm in the following section.

5.1.3 *aRNAque's EA*

As described in the introductory chapter, an EA starts with an initial population of solutions and sequentially applies the mutation and selection operators on the solutions through generation until a termination criterion is satisfied. How does `aRNAque` generate the initial population of RNA sequences? This section describes how the initial population is generated and then provides the core pseudocode of our EA.

For a given population size n and a target structure $\mathcal{S}^*$ of length L, an initial population P is generated randomly as follows:

1. Select randomly L nucleotides in $\mathcal{N}$
2. Identify the base pair position (i, j) in the random sequence, select randomly a base pair in the set of canonical base pairs $\mathcal{C}$ and fix the first nucleotide of the selected canonical base pair at the position i and the second at position j.
3. Repeat 2. for all base pair positions in the target structure
4. Repeat 1. 2. and 3. $n-$times.

Let T be the maximum number of generations and F_t the set of all sequences found at a given time t. After the initial population of RNA sequences is generated, our algorithm is described in algorithm 2. The stopping criteria are two: 1) the number of generations (t) is equal to the max number of generations (T) or 2) the minimum hamming (or base pair) distance of the best RNA sequence solution to the target is 0 (i.e the maximum fitness value is 1).

In sum, our EA relies on three objective functions and implements a Lévy flight mutation scheme. We assess the performance of our EA and the existing tools using three benchmark data sets presented in the previous chapter (Section 4.5). The following section describes the benchmark protocols applied for each data set and different RNA inverse folding tools considered in the

Algorithm 2: aRNAque' EA

/* $P' = \{\phi'_1 \dots \phi'_n\}$: the best population;
$P = \{\phi_1 \dots \phi_n\}$: the initial population of n RNA sequences;
$P_C = \{w_{AU}, w_{GU}, w_{GC}\}$: a vector containing the weights associated with each base pair;
$P_N = \{w_A, w_U, w_C, w_G\}$: a vector containing the weights associated with each nucleotide;
$\mathcal{D}$: a given probability distribution (Lévy or Binomial) with parameter p and L, where L is the length of the target RNA structure;
T: the maximum number of generations;
n: the population size ;
$f(;)$: the fitness function used. It can be the hamming, base-pair or energy distance;
σ^*: the target structure in its string representation;
$\mathcal{P}$: the energy parameters used for the folding */

Input: $n, T, P_N, P_C, P, \mathcal{D}(p, L), f(;), \sigma^*, \mathcal{P}$

Output: Best population P'

$P' \leftarrow P$; // Assign the initial population to the best population

$t \leftarrow 0$; // Initialize the number of generations to 0

while $t \leq T$ & $f(\sigma^{MFE}(\phi_b), \sigma^*) \neq 1$ **do**

  $\Sigma \leftarrow \{\arg\min_{\sigma \in \Sigma} \Delta G(\phi_i, \sigma, \mathcal{P})\}$, ; // Fold each sequence $\phi_i \in P'$ and store them in Σ. Where $i \in \{1, 2, \dots, n\}$, Γ the structural ensemble and $\Delta G(\phi_i, \sigma)$ the free energy computed using the parameters $\mathcal{P}$

  $\kappa = \lfloor (n \times 0.1) \rfloor$; // The number of RNA sequences to copy in the next generation without mutating them.

  $F \leftarrow \{f(\sigma, \sigma^*) | \forall \sigma \in \Sigma\}$; // Evaluate the fitnesses of the folded population to the target strucre σ^* and store them in a list F

  $E_\kappa \leftarrow \{\phi_1 \dots \phi_\kappa\} \sim F$; // copy of the 10% best sequence based on their fitnesses F.

  $P_S \leftarrow \{\phi_i\} \sim F$, where $i \in \{1, 2, \dots, n - \kappa\}$; // Randomly sample $(n - \kappa)$ RNA sequences from P' with respect to their fitnesses F.

  $M \leftarrow \texttt{mutate}(P_S, \mathcal{D}(p, L), P_C, P_N)$; // Mutated the selected sequences using the mutation algorithm presented in the main text in out paper.

  $P_b \leftarrow M \cup E_\kappa$; // Combine the mutated population and the best solutions to form the new population that will be evolved in the next generation

  $\phi_b \leftarrow \arg\max_{\delta \in \Sigma} f(\sigma, \sigma^*)$;

  $t \leftarrow t + 1$; // Increment the time step (the number of generations)

context of this work. Furthermore, it provides an overview of various folding tools and the configuration parameters used for the benchmark.

5.1.4 *Benchmark parameters and protocols*

For the benchmark results presented in this work, we use three datasets: the `Eterna100` dataset, `RFAM` dataset and `PseudoBase++` dataset. Depending on the datasets, a specific RNA folding tool is used. This section gives more details about `aRNAque`'s parameters, energy parameters and other tools parameters used for the benchmark results presented in this chapter.

Folding tools

Two tools for pseudoknotted RNA folding are considered in this work: `HotKnots` and `IPknot`. For pseudoknot-free RNA folding, we used `RNAfold`. For the mutation parameter and GC-content analysis presented in our work, we used `IPknot`, and both `HotKnots` and `IPknot` for `PseudobBse++` benchmarks. To be able to use `HotKnots` in `aRNAque` without copying `aRNAque` in the bin directory of `Hotknots`, we have performed some modifications on `Hotknots` source code. Details on the modifications are provided in the Section B.5. Furthermore, we considered `pkiss`, a well known tool for K-type pseudoknot prediction, but since the `PseudoBase++` dataset contains just 4 K-type pseudoknotted structures and `pKiss` has higher time complexity ($O(n^6)$), we did not find it efficient for the benchmark we performed.

Mutation parameters tuning

The main challenge for an evolutionary algorithm is to find optimum parameters such as mutation rate, population size and selection function. We used 80 pseudoknotted targets with lengths from 25 to 181 nucleotides for the mutation parameter analysis. We set the maximum number of generations T to 200 and the population size n to 100. The stopping criteria are two: 1) the number of generations (t) is equal to the max number of generations (T) or 2)the minimum hamming (or base pair) distance of the best RNA sequence solution to the target is 0. The best mutation parameters (c^* for Levy and μ^* for Binomial) are those that have the lowest median number of generations. The best mutation parameters obtained for both binomial and Lévy mutation

modes are used to benchmark and compare the results on the entire datasets of RNA structures.

Benchmark on the `PseudoBase++` dataset

Four benchmarks are performed on the pseudoknotted dataset: 1) mutation parameter analysis, 2) the GC-content and diversity analysis, 3) Local search versus Lévy search, 4) `aRNAque` (Lévy search) versus `antaRNA`. For the `aRNAque` (Binomial and Lévy) case, the four benchmarks share the same number maximum number of generations ($T = 200$), population size ($n = 100$), stopping criteria ($t = T$ or min fitness equals 0). For the `antaRNA` benchmark, the maximum number of iterations was set to 1200, and a slight modification was made to allow the support of the folding tool `HotKnots` (See Section B.5). For booth tools and each benchmark, 20 runs were launched independently in parallel on a computer with the same resources, resulting in 20 designed sequences per pseudoknotted target structure. To measure the performance of each tool, each designed sequence s is folded into a secondary structure $\mathcal{S}$ and the similarities between $\mathcal{S}$ and $\mathcal{S}^*$ are computed using the base pair distance. For the GC-content benchmark, four GC-content values are considered, $\{0.25, 0.5, 0.75, 1\}$ and the setting of each tool remains the same.

Benchmark on the `Eterna100` dataset

We performed two benchmarks are one the `Eterna100` dataset: 1) a benchmark on the `Eterna100-V1` dataset using the `Turner1999` energy parameter and the both versions of `aRNAque` (one point and Lévy mutation), 2)a benchmark on the `Eterna100-V2` dataset using the `Turner2004` energy parameter and both versions of `aRNAque` (one point and Lévy mutation). For each of the `Eterna100` benchmark we used the same evolutionary algorithm parameters; a maximum of $T = 5000$ generations (i.e. a maximum of $500,000$ evaluations), a population size of $n = 100$ and the same stopping criteria (the number of generation $t = T$ or min fitness equals 0). For both local and Lévy search, 5 runs were launched independently, which results in 5 designed sequences per target. We define success rate simply as the number of successfully designed targets. A target is considered successfully designed when at least one of the designed sequences folds into the target structure.

For the benchmarks peformed on `ERD`, `NUPACK`, and `SentRNA` the default parameters were used. For `NEMO`, the number of iteration

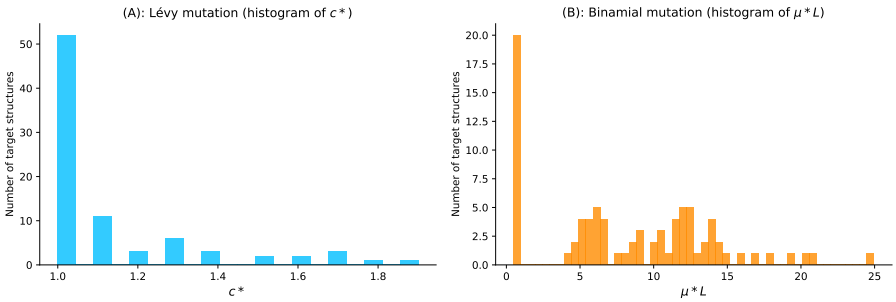


Figure 5.2: **Parameter tuning for both binomial and Lévy mutation schemes**. (A) Lévy mutation parameter tuning. Histogram of best exponent parameter (c^*) for a set of 81 target structures with different pseudoknot patterns and various lengths. The most frequent best exponent value is 1. (B) Binomial parameter tuning. Histogram of best mutation rate (μ^*) for the same set of 81 target structures with different pseudoknots and various lengths. The most frequent best parameter is the low mutation rate ($\approx 1/L$). For some structures, the best mutation rate is the high one for different lengths as well.

was set to 2500 and for `RNAinverse` the objective function was set to be the partition function and the number of iteration at 1200.

Benchmark on the `non-Eterna100` dataset

For the `non-EteRNA` dataset, only the Turner2004 energy parameters were used. The maximum number of generations was set to be 5000. The mutation parameters (P_C and P_N) were chosen to be close to the nucleotide distribution of the RNA sequence in nature [47].

5.2 EXPERIMENTAL RESULTS

As mentioned in Chapter 4, the validation of computational tools for RNA inverse folding can include in vivo or in vitro experiments. In the context of this work, only in silico experiments are used to evaluate the performance of the existing tools, including `aRNAque`. This is done through a benchmark protocol described in the previous section. This work's computational tools require an RNA secondary structure as an input target. Two RNA secondary structures are considered: the pseudoknot-free and the pseudoknotted target structures, and both are supported in `aRNAque`. Therefore, we evaluate `aRNAque` performance for different target secondary

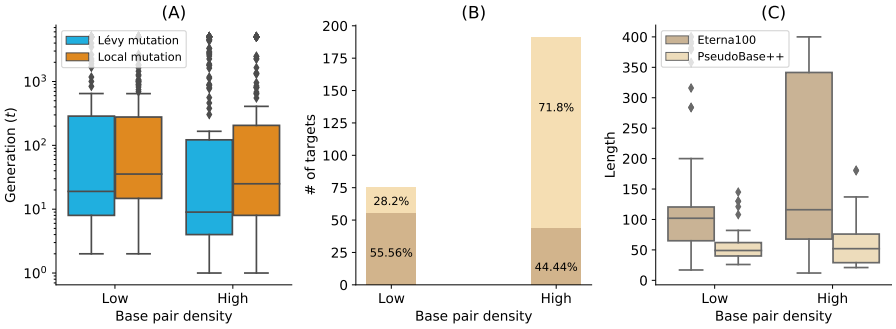


Figure 5.3: **Lévy mutation *vs.* Local mutation: performance analysis with respect to the base-pair density**. The higher the base-pair density is, the more useful the Lévy mutation scheme to speed up the optimization EA. (A) Distributions of number of generations for the low and high base-pair density targets of the `Eterna100` dataset. (B) Percentages of targets with low and high base-pair density for the `Eterna100` and `PseudBase++`. (C) The length distributions of the low and high base-pair density pseudoknot-free and pseudoknotted targets.

structures separately. Three data sets of secondary structure targets are used: the `Eterna100` and non-`Eterna100` that contain pseudoknot-free targets, and the `PseudoBase++` which contains only pseudoknotted targets. Using the three data sets, this section presents the experimental results concerning the quality of the designed RNA sequences (i.e. the GC-content and diversity or positional entropy), the CPU required and the success rate of each tool considered when benchmarked using a specific data set.

5.2.1 *aRNAque's performance on pseudoknot-free target structures*

We compared the performance of `aRNAque` for pseudoknot-free target using the benchmark datasets: the non-`Eterna100` and the `Eterna100`. This subsection presents the statistical results obtained compared to benchmarked existing tools and the results found in the literature. In addition, we compared the performance of `aRNAque` (Lévy mutation) to the one of Ivry et al. [88] on a tripod pseudoknot-free RNA secondary structure.

5.2.1.1 *Performance on `Eterna100` dataset*

A first benchmark was performed on the `Eterna100` datasets. First, on the `Eterna100-V1` dataset, the Lévy flight version of `aRNAque`

Table 5.1: **Summary of performance of aRNAque vs the 7 other algorithms benchmarked on EteRNA100-V1** by Anderson-Lee et al. [3] (using the recent energy parameter sets, the Turner2004)

Methods	Number of puzzles solved
aRNAque	**72/100**
RNAinverse	66/100
Learna	66/100
ERD	65/100
SentRNA, NN + full moveset	60/100
MODENA	54/100
NEMO	50/100
INFO-RNA	50/100
NUPACK	48/100
DSS-Opt	47/100
RNA-SSD	27/100

successfully designed 89% of the targets and the one-point mutation (local mutation) version achieved 91% of success, suggesting that for some target structures, local mutation can outperform the Lévy mutation scheme. Combining the two solutions, aRNAque solved in total 92% of the targets of Eterna100-V1.

When analysing the performance of Lévy flight for low and high base pair densities separately, the median number of generations of high base pair density targets was lower than the one with low base-pair density (8 generations for high density and 18 for the low base pairs density targets). The same observation was drawn for the success rate. For the low base-pair density targets, the Lévy flight achieved 87% (49/56) success whereas, for the high base-pair density, it achieved 91% (40/44). The same analysis can be done when comparing the one-point mutation results for the high-density targets to the Lévy flight mutation. The median number of generations for the low-density targets when using a one-point mutation operator was 34 (respectively 24 for the high base pair density targets) (see Figure 5.3A).

Another benchmark was performed on Eterna100-V2 with aRNAque achieving a 93% success rate when combining the designed solutions for both mutation schemes. Compared to recently reported

Table 5.2: **Summary of performance of aRNAque vs the 10 other algorithms benchmarked on the non-EteRNA100** by Anderson-Lee et al. [3]

Methods	Number of puzzles solved
SentRNA, NN + full moveset	57/63
ERD	54/63
SentRNA, NN + GC pairing	53/63
SentRNA, NN + All pairing	53/63
aRNAque	**52/63**
RNA-SSD	47/63
SentRNA, NN only	46/63
INFO-RNA	45/63
MODENA	32/63
NUPACK	29/63
IncaRNAtion	28/63
Frnakenstein	27/63
RNAinverse	20/63
RNAfbinv	0/63

benchmark results [106], aRNAque achieved almost similar performance to NEMO on Eterna-V2: one target was unsolved by all existing tools and one target solved only by NEMO remained unsolved by aRNAque, outperforming all existing EA methods.

For the robustness analysis, Table 5.1 presents the benchmark results on Eterna100-V1 using the Turner2004 energy parameters sets. It shows that the evolution algorithm we propose can solve $\approx$ 72% of the dataset, and it surpasses the 4 methods we benchmarked and all the tools already benchmarked in [171]. We can also solve approximately 23 targets more than NUPACK, which is also minimizing the ensemble defect and that shows the importance of a population-based algorithm. Compared to the existing EA-based algorithms, our EA can solve approximately 18 targets more than MODENA and 7 targets more than ERD.

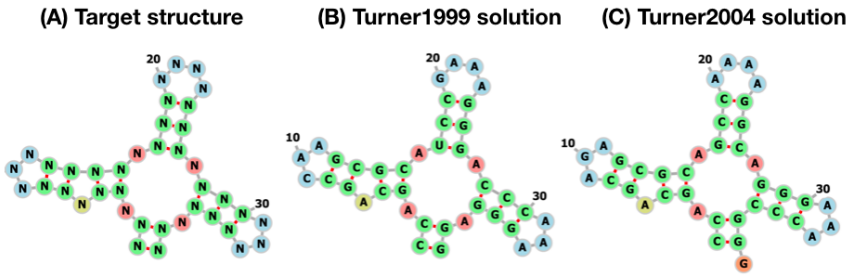


Figure 5.4: **aRNAque's performance on a TRIPOD secondary structure**. (A) The tripod target structure. (B) aRNAque's solution using the Turner1999 energy parameter sets. (C) aRNAque's solution using the Turner2004 energy parameter sets.

5.2.1.2 *Perfomance on non-Eterna100*

Additionally to the Eterna100 dataset, we also used the non-EteRNA dataset collected from the RFAM database to assess the aRNAque's performance on pseudoknot-free target secondary structure. Compared to other tools, the statistical results are presented in Table 5.2.

The results show that our method surpasses 8/10 of other tools. ERD solved 2 more targets than our method because of its strong decomposition capacity, which allows it to solve the entire **dataset B**. With the advantage that our evolutionary algorithm also allows us to fit the nucleotide distribution parameters taken from natural RNA directly in the mutation parameters, we can solve 21/24 targets from the **dataset B**. For the **dataset A** aRNAque solves 24/29 targets which means 2 more than the existing tools and for the 10 last targets, it solves 7 targets. Adding all these solved targets together, we obtain a result of 52/63 as presented in Table 5.2.

5.2.1.3 *aRNAque performance on a tripod secondary structure*

Finally, we performed a benchmark on a tripod target secondary structure. The tripod secondary structure was used as a third test case in the work of Ivry et al. [88], and it does not contain any pseudoknot interactions. It comprises four stems, three of which with terminal hairpins, surrounding a multibranch loop (See Figure 5.4A). The tripod target structure was proved to be very challenging, especially because of its multiloop component, which is also found in some of the unsolved Eterna100 target structures. We perform here, for both energy parameters Turner1999

and Turner2004, 100 independent designs, using a population size of 100 RNA sequences and a maximum of 5000 generations. The mutation parameters used are: $P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$, $P_{\mathcal{N}} = \{0.7, 0.1, 0.1, 0.1\}$ and $c = 1.5$. When using the Turner2004 energy parameter set, none of the 100 designed RNA sequences was successful (i.e, 0 sequence folds exactly into the target structure after 5000 generations). Figure 5.4B shows one of the best solutions obtained out of 100 designed sequences when using the Turner2004, the designed sequence folds into a structure at one error base-pair distance from the target structure. In contrast, when using the Turner1999 energy parameters, we successfully designed the tripod secondary structure (See Figure 5.4C). The 100 sequences designed folded exactly into the target structure with an average median number of generations 20. When comparing both solutions to the one obtained in [88], aRNAque (with no need of changing the RNA structure distance) can successfully design the multibranch loop component with one base pair error using the Turner2004 energy parameter whereas RNAinverse (with the DoPloCompare distance) failed to design the multibranch loop, and the solution was at 2 base-pair distance error.

5.2.2 *aRNAque's performance on pseudoknotted target structures*

Secondly, we assessed the performance of aRNAque in designing pseudoknotted target secondary structures through intensive benchmark on PseudoBase++ dataset. We then compared the results obtained to the one of antaRNA, using both folding tools Hotknots and IPknot. Furthermore, a comparison between local and Lévy mutations is provided.

5.2.2.1 *Best mutation parameter analysis on PseudoBase++: Levy mutation* vs. *local mutation*

The advantage of using a Lévy mutation is its capacity to allow simultaneous search at all scales over the landscape. The search at different scale is often dictated by the exponent parameter of the heavy-tailed distribution. In this first subsection, we analyse for 80 pseudoknotted target structures and for both mutation schemes the distributions of the best mutation parameters.

- Binomial mutation: From Figure 5.1B, the critical range was identified to be from 0 to 0.2 and as μ becomes greater than 0.1, the success rate decreases and the average number

of generations increases. For each of the 80 target structures with pseudoknots, 20 sequences were designed for $\mu \in [0, 0.2]$ with a step size of $1/L$. Figure 5.2B shows the histogram of the best mutation rate found for each target structure. Two main regimes are apparent: one where the best mutation rate is very low mutation rate ($\approx 1/L$) and another where the high mutation rate is optimal.

- Lévy mutation: From Figure 5.1C, the critical range of c was identified to be $[1, 2]$. For $c \in [1, 2]$ and a step size of 0.1, an optimum exponent parameter c^* was investigated for all the 80 target structures. Figure 5.2A shows the histogram of c^*. Contrary to binomial mutation, the optimum exponent parameter does not vary too much ($\forall \mathcal{S}, c^* \approx 1$).

Figure 5.2 shows that when using a Lévy mutation, the optimal mutation rate is the same for most target structures. In contrast, the optimum binomial mutation rate parameter μ^* mostly varies with different targets. Although both mutation schemes (for the best mutation parameters) have approximately the same success rates, the Lévy flight mutation scheme is more robust to different targets.

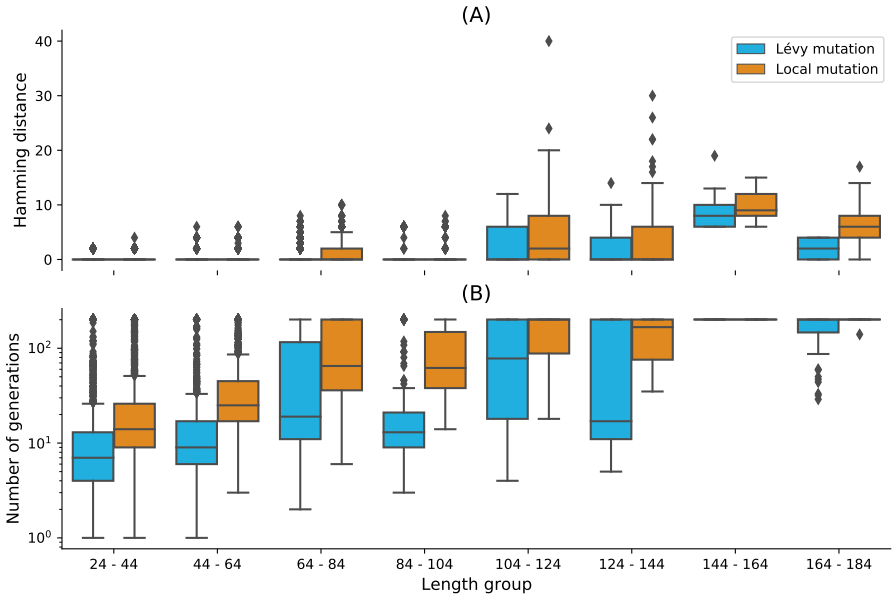


Figure 5.5: **Lévy mutation mode *vs* local mutation (one-point mutation)**. (A) Hamming distance distributions *vs.* target structure lengths. (B) Number of generations distributions for different length groups. In both (A) and (B), lower values indicate better performance. The target structures are solvable in less than 100 generations for both mutation schemes and most length groups. Still, the difference in the number of generations gets more significant as the target lengths increase, except for the two last length groups for which both mutation schemes mostly failed. The highest difference in terms of median number of generations is 150 for target lengths in the range $[124-144]$ (respectively $123, 49, 46, 16, 7, 0, 0$ for the length ranges $[84-104]$, $[64-84]$, $[104-124]$, $[44-64]$, $[24-44]$, $[144-164]$, $[164-184]$). Averaging over all length groups, the median number of generations difference between the Levy mutation and the one point mutation is 48 generations.

5.2.2.2 *Performance on* `PseudoBase++`*: Levy mutation* vs. *local mutation*

Figure 5.5 shows box plots for the base pair distance (Hamming distance) and the number of generations for increasing target lengths under our two mutation schemes: binomial at low mutation rate (or one point mutation) and the Lévy mutation. For each pseudoknotted RNA target structure in the `PseudoBase++` dataset, we designed 20 sequences. The results show that using the Lévy mutation instead of a local mutation scheme can significantly increase the performance of `aRNAque`. The gain was less significant

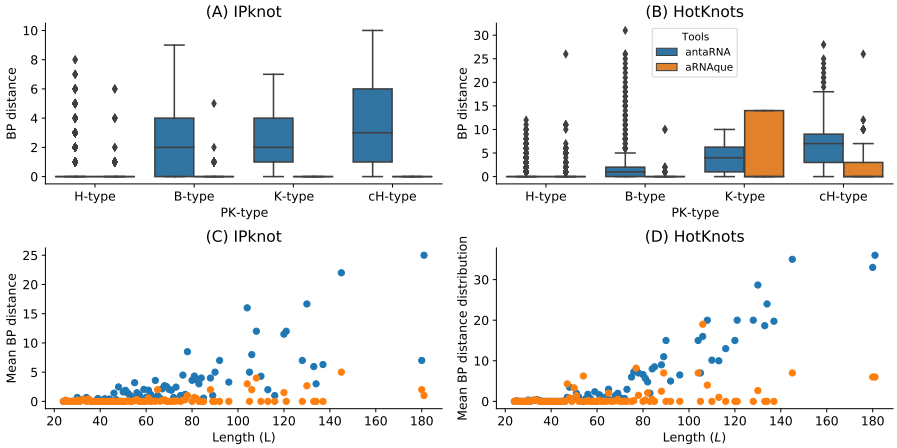


Figure 5.6: **aRNAque *vs* antaRNA on PseudoBase++ dataset using both IPknot and HotKnots**. Lower values imply better performance. (A, B) Base pair distance distributions of the designed sequences to the target structure for different pseudoknot types. (C,D) Mean base pair distance against target lengths.

in terms of designed sequences quality (base pair distance distributions, with a t-value ≈ -1.04 and p-value ≈ 0.16) but more significant in terms of the average number of generations needed for successful matches to target structures (with a t-value ≈ -3.6 and p-value ≈ 0.0004). This result demonstrates a substantial gain in computational time when using a Lévy mutation scheme instead of a purely local mutation.

5.2.2.3 *Performance on PseudoBase++: aRNAque vs. antaRNA*

We also compared the sequences designed using aRNAque (with the Lévy mutation scheme) to those produced by antaRNA. Figure 5.6A and Figure 5.6C show the base pair distance distribution for each category of pseudoknotted target structure and the mean of the base pair distance plotted against the length of the target secondary structures. For antaRNA, and when using IPknot as a folding tool, finding sequences that fold into the target becomes increasingly difficult with pseudoknot complexity (median base-pair distance distribution increases). On the other hand, aRNAque's performance improves as pseudoknot complexity increases (e.g. the mean base-distance decreases with the pseudoknot complexity).

A second benchmark using HotKnots as a folding tool was performed on the same dataset. For both aRNAque and antaRNA, the

more complex the pseudoknot motifs, the worse is the tool performance (median of the base-pair distance distribution increases). Figure 5.6B and Figure 5.6D show the base pair distance distributions with respect to the pseudoknot motifs for both `aRNAque` and `antaRNA`. Even though both performances degrade as target length increases, `aRNAque` (Lévy flight evolutionary search) performance remains almost constant for all the target lengths greater than 60.

5.2.3 *Quality of the designed RNA sequences*

In addition to the successful rate analysis, we assessed the quality of the designed RNA sequences by analysing both GC-content and diversity of the pseudoknotted dataset using `IPknot`. This section presents the results obtained and a comparison to `antaRNA` designed sequences.

5.2.3.1 *GC–content analysis of the designed sequences using `IPknot`*

The GC–content of an RNA sequence S measures the concentration of G-C nucleotide in S and influences its stability and biological function. Therefore, the ability of an inverse folding tool to control the GC–content is of vital importance for designing functional RNA sequences. Both `antaRNA` and `aRNAque` allow to control the GC–content at different levels of the optimization process: `aRNAque` through the mutation parameters P_C and P_N; `antaRNA` with the parameter $tGC \in [0, 1]$. In this section, we compare the performance of each tool for fixed GC–content values and analyse each tool's ability to control the GC–content. For each pseudoknotted target structure in the `PseudoBase++` dataset, four different GC-content values $\{0.25, 0.5, 0.75, 1\}$, a poll of 20 sequences is designed using `IPknot` as folding tool. That results in 5320 designed sequences for each GC-content value and tool. The number of successes is the total number of sequences that fold exactly into the given target structure (i.e. the designed sequence folds into a structure at base-pair distance 0 from the target structure). Figure 5.7 shows respectively the base pair distance distributions, the GC distance distributions and the number of successes for both `aRNAque` and `antaRNA`. The results show that the performance (in terms of success number) varies considerably with the GC–content values for both tools, and the best performance is obtained for both tools with a GC–content value of 0.5. When comparing the GC-content distance (i.e absolute

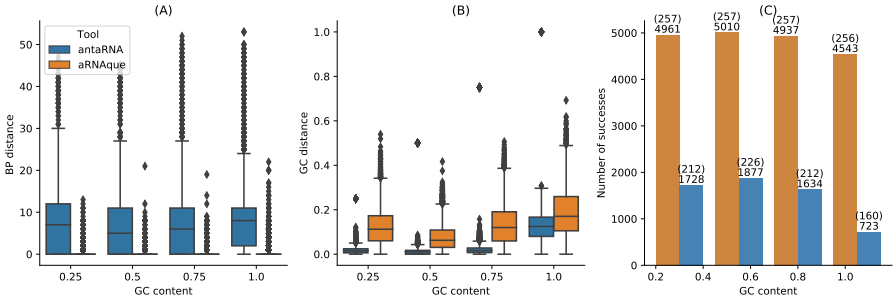


Figure 5.7: **aRNAque *vs* antaRNA on PseudoBase++ dataset using IPknot: GC–content analysis**. (A) Base-pair distance ditributions. (B) GC–content distance distributions. The difference betwen the targeted GC-content and the actual GC-content values. In (A,B), lower values imply better performance. (C) Number of successes realised by both inverse folding tools. Two values are considered: the up value represent the number targets successfully solved for each GC-content value out of the 266 targets benchmarked; the down values represent the number sequences folding into the targeted secondary structure.

value of the difference between the targeted GC–content and the actual GC–content values of the designed sequences) distributions, both GC–content distance median distributions increase, whereas antaRNA controls significantly better the GC-content (See Figure 5.7B). On average, for the respective GC-content values $\{0.25, 0.5, 0.75, 1\}$, antaRNA's sequences have respectively $0.2569, 0.4952, 0.7314, 0.8684$ whereas aRNAque's sequences have respectively $0.3649, 0.4910, 0.6231, 0.811$; the main difference is at fixed GC-content values 0.25 and 0.75. Even though antaRNA designs sequences with better control of the GC-content, the gap in success rate still remains remarkable compared to aRNAque(See Figure 5.7A and Figure 5.7C).

5.2.3.2 *Diversity of the designed sequences*

Another advantage of using a Levy mutation when designing RNA sequences is to increase the chance of designing sequences with high diversity. Here, we use the positional entropy of each pool of 20 sequences previously designed for each pseudoknotted target structure to compare the diversity of RNA of both tools antaRNA and aRNAque (Lévy search). We also compare it to the diversity of the designed sequences using the old version of aRNAque (Local search). The results show that the sequence diversity of

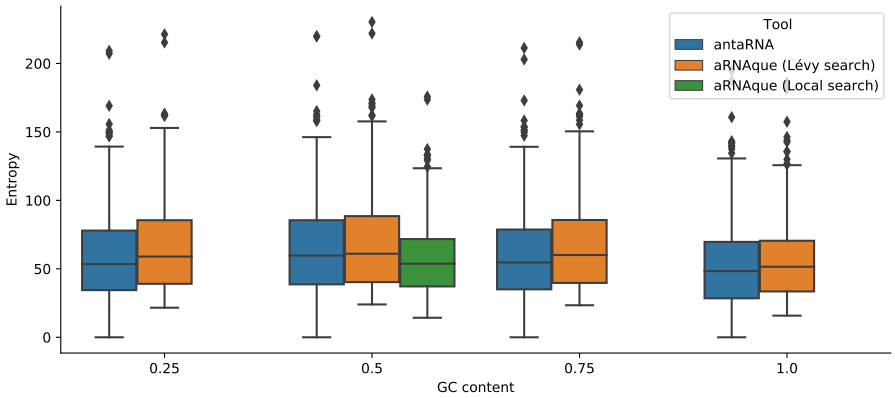


Figure 5.8: **aRNAque *vs* antaRNA on PseudoBase++ dataset using IPknot: Diversity analysis**. The positional entropy distributions plotted agains the targeted GC–content values. Higher values imply better performance.

both antaRNA and aRNAque (Lévy search) varies with the GC–content values, where the more diversified pool of sequence is achieved with a GC–content value of 0.5. When comparing the pool of designed sequences with highest entropy (i.e. with a fixed GC-content of 0.5) to the one of the old version of aRNAque (Local search), the aRNAque (Lévy search) and antaRNA produce sequences with similar entropy (i.e. with a median entropy of 61.01 for Lévy search respectively 59.65 for antaRNA (see Figure 5.8), whereas the entropy of the sequences designed using the Local search is lower. For the three others fixed GC-content values (i.e. $\{0.25, 0.75, 1\}$, aRNAque (Lévy search) produces sequences with the highest entropy (respectively a median entropy of $58.9, 60.08, 51.52$ against $53.42, 54.63, 48.38$ for antaRNA).

5.2.4 *Complexity and CPU time comparison*

We finally analysed the design performance of aRNAque relatively to the CPU time needed. This section presents aRNAque statistical results compared to two main tools: RNAinverse for the pseudoknot-free targets and antaRNA for the pseudknotted targets.

5.2.4.1 *CPU time* vs. *success rate using RNAfold: RNAinverse vs. aRNAque on EteRNA100-V1.*

Since our previous benchmarks on EteRNA100-V1 using the Turner2004 energy's parameters reveal that RNAinverse, one of the oldest in-

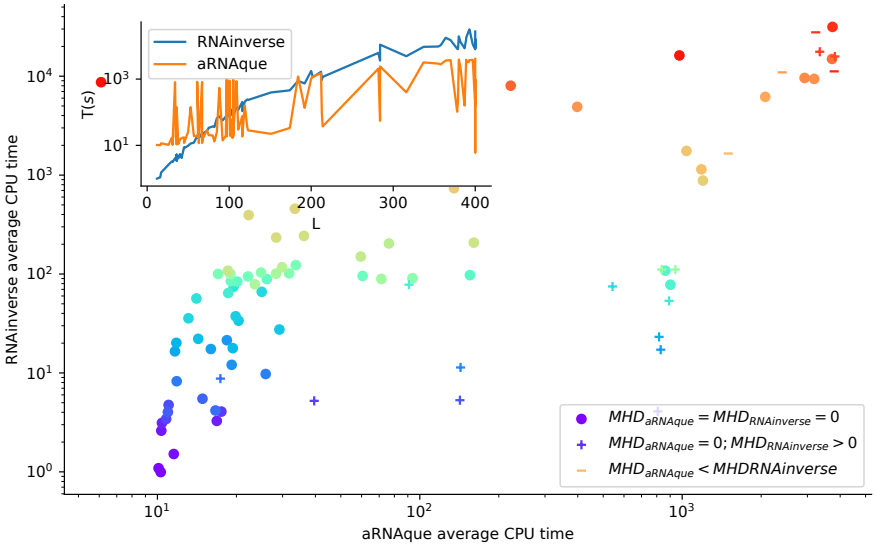


Figure 5.9: CPU **time: `RNAinverse` *vs.* `aRNAque`**. Each bubble corresponds to a target structure in `EteRNA100` dataset and, their colours are proportional to the length of the targets. In the legend, MHD stands for Median Hamming distance, and the different markers represent—('o') 100% success for both tools—('+') 100% success for `aRNAque` and not for `RNAinverse`—('-') for the case both tools fail to find at least one sequence that folds into the target. Underlying the CPU time difference is the inside plot that shows the CPU time (in seconds) as a target length function.

verse folding tools, stands behind `aRNAque` solving 66% of the dataset; we have chosen to compare its computational time to our implementation (See Table 5.1). The inset of Figure 5.9 shows the CPU time in seconds needed to design for each target in the `EteRNA100-V1`, 5 sequences. As the `RNAinverse` time increases exponentially with the length of the target, the `aRNAque` one does not.

When comparing the ratio between the success rate and CPU time, `aRNAque` mostly succeeded in finding at least one sequence that folds to the target with lower CPU time costs for average target lengths. In contrast, `RNAinverse` accuracy is lower, and the CPU time is expensive. The increase in CPU time may be because of the use of the partition function as the objective function.

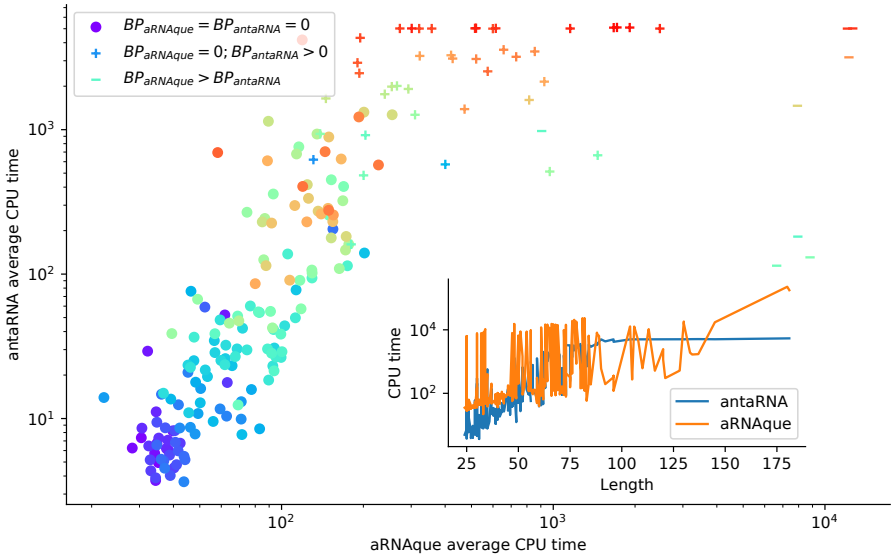


Figure 5.10: **CPU time analysis using `Hotknots`: `antaRNA` *vs.* `aRNAque`.** Each bubble corresponds to a target structure in `PseudoBase++` dataset and, their colours are proportional to the length of the targets. In the legend, BP stands for Median base pair distance, and the different markers represent—('o') 100% success for both tools—('+') 100% success for `aRNAque` and not for `antaRNA`—('-') for the case `aRNAque`'s desinged sequences are of median base pair distances greater than the one of `antaRNA`. Underlying the CPU time difference is the inside plot that shows the CPU time (in seconds) with respect to the target length.

5.2.4.2 *CPU time* vs. *success rate using `Hotknots`: `antaRNA` vs. `aRNAque` on `PseudoBase++`*

We also compare `aRNAque`'s computational time to the one of `antaRNA`. For both tools, 20 sequences were designed for each target structure of the `PseudoBase++` dataset. The GC–content value used for both tools is 0.5, and the maximum number of interactions for `antaRNA` is 5000. Figure 5.10 shows the median CPU time of the 20 runs in seconds for both tools plotted against each other. We analysed the CPU time by partitioning the data into three groups: 1) a set for which both tools have a median base-pair distance of 0 (158 entries marked with o); 2) another set for which `aRNAque` has a median base-pair distance is 0 and `antaRNA` (41 entries marked with +); 3) the last set for which `antaRNA` designs are of better quality (9 entries mark as -). For the first group, we can notice that for most targets of short length `antaRNA` is faster than `aRNAque`. For the second group, although

antaRNA average CPU time remains smaller, aRNAque's success rate outperformed antaRNA. On the one hand, aRNAque average CPU time is higher than the one of antaRNA, but this could be due to its population-based algorithm, which often allows for designing more successful sequences. On the other hand, antaRNA is faster but less successful. Increasing antaRNA's number of iterations will indeed increase the CPU time, but it may improve the quality of the designed sequences.

5.3 CONCLUSION

In this work, we investigated an evolutionary approach to improve the existing solutions to the RNA inverse folding problem. As a result, we proposed a new EA python tool called aRNAque. aRNAque implements a Lévy flight mutation scheme and supports pseudoknottted RNA secondary structures. The benefit of a Lévy flight over a purely local mutation search allowed us to explore RNA sequence space at all scales. Such a heavy-tailed distribution in the number of point mutations permitted the design of more diversified sequences, avoiding the pitfalls of getting trapped in a local optimum.

Our results show general and significant improvements in the design of RNA secondary structures compared to the standard evolutionary algorithm mutation scheme with a mutation parameter $\approx 1/L$, where L is the sequence solution length. Lévy flight mutations lead to a greater diversity of RNA sequence solutions and, in many cases, reduce the evolutionary algorithm's number of evaluations, thus improving computing time.

Part III

GENERAL CONCLUSION AND DISCUSSIONS

6

ADVANTAGES AND LIMITATIONS OF THE PROPOSED METHODS

In the presented thesis, we have summarized some molecular background and biological functions of nucleic acids, especially ncRNAs. Because of the implication of the secondary structure of ncRNAs in performing biological functions and the separation of the folding time scale, our study focuses on the secondary structure of ncRNAs. Therefore, we have introduced the concepts of RNA bioinformatics and the essential computational problems related to the secondary structure of ncRNAs, such as RNA folding and the inverse folding. We presented a comprehensive literature review on existing tools that deal with both problems and some limitations for each tool. Despite advanced field results, we have introduced two new computation tools: `RAFFT` and `aRNAque`. What are the advantages and limitations of those tools? Is there any room for further improvements? How do these tools relate to evolutionary dynamics? In this concluding chapter, we will try to provide an answer to these questions by first discussing the advantages and the limitations of the tools previously introduced.

6.1 `rafft`: LIMITATIONS AND FUTURE WORKS

We presented in Chapter 3, `RAFFT`, a computational tool that efficiently predicts RNA folding pathways. `RAFFT` takes advantage of the FFT to reduce its mean computational time to $O(N^2)$, especially for long RNA sequences (length $\geq 10^3$). We assessed `RAFFT` performance for both the secondary structure prediction task and the RNA kinetics. In both cases, `RAFFT` shows important improvements. However, `RAFFT` also presents some limitations that will be addressed in the following section. We also discuss in this section some further improvements and applications.

To first assess `RAFFT` performance for the folding task, two structure estimates were compared with our method: the thermodynamic-based tools computed using `RNAfold`, `LinearFold`, `RNAstructure` and the ML estimate using `MxFold2` and `CONTRAfold`. When we considered the lowest energy structure, the comparison of `RAFFT` to existing tools confirmed the overall validity of our approach. In more detail, a comparison with thermodynamic/ML models

yielded the following results. First, the ML predictions performed consistently better than both RAFFT and other approaches, where the PPV = 70.4% and sensitivity = 77.1% on average. Second, the ML methods produced loops, such as long hairpins or external loops. We argue that the density of those loops correlates with the ones in the benchmark dataset, which a PCA analysis revealed too.

In contrast, the density of similar loops was lower in the structure spaces produced by RAFFT and other thermodynamic-based methods, implying some over-fitting in the ML model. Finally, known structures obtained through covariation analysis reflect *in vivo* structure conditions. Therefore, the structures predicted by ML methods may result from their sequences alone and their molecular environment, e.g. chaperones. We expect the thermodynamic methods to provide a more robust framework for studying sequence-to-structure relations. Concerning thermodynamic-based tools, we obtained a substantial gain of performance when analyzing $N = 50$ predicted structures per sequence, not only the lowest energy one. This gain was even more remarkable for sequences with fewer than 200 nucleotides, reaching the accuracy of ML predictions.

So how does RAFFT predictions contain structures that are more relevant than the MFE, although these structures are less thermodynamically stable? The interplay of three effects may explain this finding. First, the MFE structure may not be relevant because active structures can be in kinetic traps. Second, RAFFT forms a set of pathways that cover the free energy landscape until they reach local minima, yielding multiple long-lived structures accessible from the unfolded state. Third, the energy function is not perfect, so that the MFE structures computed by minimizing it may not in fact be the most stable.

We also showed that the fast-folding graph produced by RAFFT can be used to reproduce state-of-the-art kinetics, at least qualitatively. Our method demonstrated three main benefits. First, the kinetics can be drawn from as few as 68 structures, whereas the barrier tree may require millions. Second, the kinetics ansatz describes the complete folding mechanism starting from the unfolded state. Third, for the length range tested here, the procedure did not require any additional coarse-graining into basins. (Longer RNAs might require such a coarse-graining step, in which structures connected in the fast-folding graph are merged together).

Based on our results, we believe that the proposed method is a robust heuristic for structure prediction and folding dynamics. The folding landscape depicted by RAFFT was designed to follow the kinetic partitioning mechanism, where multiple folding pathways span the folding landscape. This approach has shown good predictive potential. Furthermore, we derived a kinetic ansatz from the fast-folding graph to model the slow part of the folding dynamics. It was shown to approximate the usual kinetics framework qualitatively, although using many fewer structures.

However, further improvements and extensions of the algorithm may be investigated. First, the choice of stems is limited to the largest in each positional lag, a greedy choice which may not be optimal. Second, we have constructed parallel pathways leading to diverse, accessible structures. Still, we have not given any thermodynamic-based criterion to identify which are more likely to resemble the native structure. We suggest using an ML-optimized score to this effect.

Our method can also find applications in RNA design, where the design procedure could start with identifying long-lived intermediates and using them as target structures. We also believe that mirror encoding can be helpful in phylogenetic analysis. Indeed, the correlation spectra $\mathrm{cor}(k)$ computed here contained global information of base-pairing that can be used as a similarity measure.

Finally, the versatile method implemented in RAFFT gives possibilities for an alternative application of the FFT in RNA-RNA interaction. The underlying idea is that instead of encoding a sequence X and its mirror sequence $\bar{X}$, one can consider two encoded sequences X and Y, and the correlation between them will allow identifying the fraction of high interaction between two RNA sequences quickly. In general, RNA-RNA interaction prediction methods are divided into three groups: alignment like methods, MFE methods and comparative methods. MFE methods constitute the majority of the RNA-RNA interaction tools, with the only difference often based on whether the method considers intramolecular interactions. Some methods measure the accessibility of binding region (Intra and inter interactions) [7, 33, 199]. We suggest neglecting intramolecular interactions and intermolecular binding pairs for a preliminary implementation.

In sum, RAFFT provides a versatile framework in which the kinetic partitioning mechanism can be simulated. Therefore, it allows for predicting an ensemble of concurrent RNA folding pathways ending in different metastable conformations. This

result contrasts traditional thermodynamics techniques that find a single MFE structure. However, further improvements of RAFFT could be investigated:

- the limitation of the choice of stems to the largest one in each positional lag is a greedy choice that may not be optimal. We propose to add stochastic noises in the choice of positional lag to keep, such that running multiple times RAFFT, one can overcome some greediness bottlenecks.
- our method constructs parallel pathways leading to a diverse set of accessible structures. Still, we have not given any thermodynamic-based criterion to identify which are more likely to resemble the native structure. We suggest using an ML-optimized score to investigate the restrained ensemble of structures predicted by RAFFT.
- structures connected in the parallel pathways are separated by the formation or unfolding of a single stem. As mentioned above, RAFFT does not account for barriers between structures that stem formation could involve. Therefore, we propose to apply a post-treatment on the folding graph, where the folding path between structures is investigated using the set of valid atomic folding moves (*e.g.* individual base-pair formation).

In addition to these possible improvements, we presented two possible applications: RNA design and RNA-RNA interactions. In Section 6.3, we discuss another application in the study of evolutionary dynamics.

6.2 arnaque: LIMITATIONS AND PERSPECTIVES

We have provided in Chapter 5, a new tool aRNAque, implementing an EA with a Lévy flight mutation scheme that supports pseudoknottted RNA secondary structures. We discuss in this section the advantages of using aRNAque for RNA design and some limitations that could be addressed for further improvements.

The Lévy mutation scheme offered exploration at different scales (mostly local search combined with rare big jumps). Such a scheme significantly improved the number of evaluations needed to hit the target structure, while better avoiding getting trapped in local optima. The benefit of a Lévy flight over a purely local mutation search allowed us to explore RNA sequence space at all

scales. Such a heavy tailed distribution in the number of point mutations permitted the design of more diversified sequences. The main advantage of using a Lévy flight over local search was more remarkable for the pseudoknotted RNA targets, which is a reduction in the number of generations required to reach a target (see Figure 5.5). This is because the infrequent occurrence of a high number of mutations allow a diverse set of sequences among early generations, without the loss of robust local search. One consequence is a rapid increase in the population mean fitness over time and a rapid convergence to the target of the maximally fit sequence. To illustrate that advantage, we ran `aRNAque` starting from an initial population of unfolded sequences, both for a "one point mutation" and "Lévy mutation".

Figure 6.1A and Figure 6.1B show respectively the max/mean fitness over time and the number of distinct structures discovered over time plotted against the number of distinct sequences. When using a Lévy mutation scheme, the mean fitness increases faster in the beginning but stays lower than that using local mutations. Later in the optimisation, a big jump or high mutation on the RNA sequences produces structures with fewer similarities and, by consequence, worse fitness. In the $(5-10)^{th}$ generation, sequences folding into the target are already present in the Lévy flight population, but only at the 30^{th} generation are similar sequences present in the local search population. The Lévy flight also allows exploration of both the structure and sequence spaces, providing a higher diversity of structures for any given set of sequences (Figure 6.1B). Using the mean entropy of structures as an alternate measure of diversity, we see in Figure 6.1C and Figure 6.1D how a Lévy flight achieves high diversity early in implementation, and maintains a higher diversity over all generations than a local search algorithm. Although the mutation parameters P_C and P_N influence the absolute diversity of the designed sequences, the Lévy flight always tends to achieve a higher relative diversity than local search, all else being equal.

We argue that the improved performance of the Lévy mutation over local search in target RNA structures is due to the high base pair density of pseudoknotted structures. Given that pseudoknotted RNA structures present a higher density of interactions, there are dramatic increases in possible incorrect folds and thus increasing risk of becoming trapped near local optima [74]. Large numbers of mutations in paired positions, as implied by a heavy tailed distribution, are necessary to explore radically different solutions.

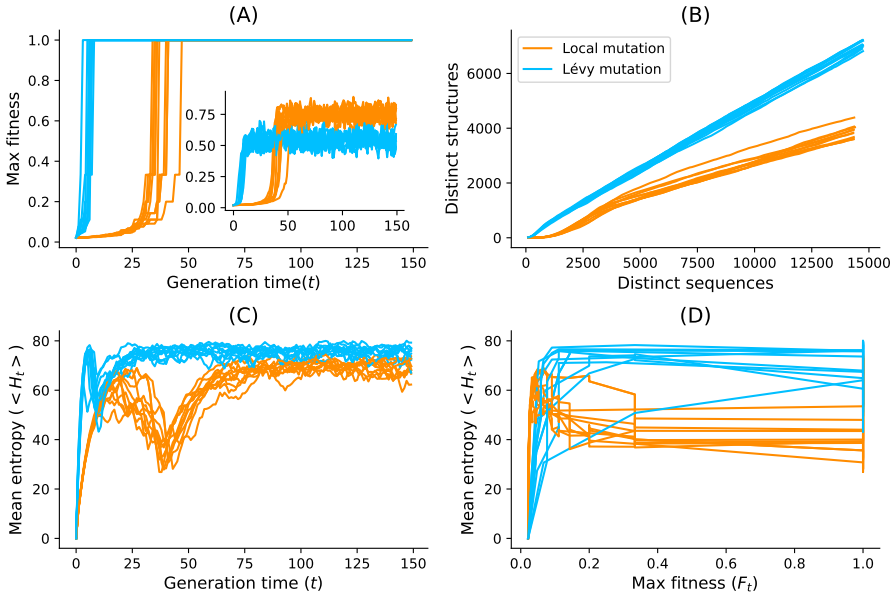


Figure 6.1: **Lévy mutation *vs* one-point mutation**. For the `Eterna100` target structure [*CloudBeta*] *5 Adjacent Stack Multi-Branch Loop*, ten independent runs were performed in which a minimum of 10 sequences were designed per run. (A) Max fitness and mean fitness (inset) over time. (B) Distinct sequences *vs.* Distinct structures over time. (C) Mean Shannon entropy of the population sequences over time for both binomial and Lévy mutation. (D) The max fitness plotted against the entropy over time.

To illustrate that Lévy flight performance could be due to base pair density, we clustered the benchmark datasets into two classes: one cluster for target structures with low base pair density (density ≤ 0.5) and a second cluster for structures with high base pair density (density > 0.5). Figure 5.3B showed the number of target sequences available in each low and high density category. The number of targets available in each category are colored according to the percentage of pseudoknot-free targets (`Eterna100-V1`) vs. targets with pseudoknots (`Pseudobase++`), showing that pseudoknots are strongly associated with high base pair densities: 71% of the pseudoknotted target structures have a high base pair density. In contrast, the `Eterna100` dataset without pseudoknots has somewhat higher representation at low base pair density. If it is true that improved Lévy flight performance is indeed tied to base pair density, it is possible that similar heavy-tailed mutation schemes could offer a scalable solution to even more complex inverse folding problems. Another measure of difficulty is the length of the target RNA secondary structure. When analysing the

mean length of the pseudoknot-free targets, the high base-pair density targets are on average 181 nucleotides longer, and the low-density base-pair targets are 139 nucleotides (See Figure 5.3C). We have 49 nucleotides for low-density targets for the pseudo-knotted targets and 52 nucleotides for the high-density targets. That suggests that the Lévy mutation may be a good standard for designing more challenging target structures.

A further effort have been made to understand the cases in which the Levy flight mutation can outperform the Binomial with low mutation rate or a constant one-point mutation rate. The key point of a Lévy mutation for the Inverse folding problem partially may rely on the base-pair density and the stability of stems with budge.

Although we believe that Lévy flight-type search algorithms offer a valuable alternative to local search, we emphasise that its enhanced performance over say `antaRNA` is partially influenced by the specific capabilities of existing folding tools. Their limitations may account for the degradation of these tools as the pseudoknot motifs get increasingly complex (i.e. the incapacity of existing folding tools to predict some pseudoknot motifs influences the performance of both `aRNAque` and `antaRNA`). The Lévy mutation has also shown less potential in controlling the GC–content of the designed sequence when compared to `antaRNA` on pseudo-knotted target structures. `antaRNA`'s parameters used in this work were tuned using `pKiss`; therefore, it could be possible room for improving the benchmark presented here by retuning them using `IPKnot` or `HotKnots`. Another possible limitation is the fact that most target structures were relatively easy to solve (in less than 100 generations), which possibly allowed local search to perform better than Lévy search in some cases. Further research on more challenging target structures will improve our understanding of which conditions favour local *vs.* Lévy search.

6.3 rafft, arnaque and evolutionary dynamics perspectives

The RNA inverse folding has deep connections with theoretical evolutionary dynamics studies, where the sequence-secondary structure relationship is a popular model for studying the genotype/phenotype maps [66, 89]. The folding tool usually maps each sequence to a secondary structure, e.g. `RAFFT` pathways could be used to compute developmental paths from sequence to secondary structure and then use the most dominant structure as

the phenotype realization of a genotype RNA sequence. Therefore, the two tools we previously introduced have a direct connection with the evolutionary dynamic, where `aRNAque` simulates the dynamic evolutionary process and `RAFFT` computes the genotype-/phenotype mapping. This section presents some evolutionary dynamics concepts that could be furtheer study using `RAFFT` and `aRNAque`.

Similar to EAs, implemented in `aRNAque`, simulating a dynamic evolutionary process using RNA sequence-secondary structure relationship as a model often involves a population of RNA sequences to a given target secondary structure. In such a simulation, we need three main ingredients: replication, selection and mutation. These are the fundamental and defining principles of biological systems. The underlying idea is that the genomic material (the blueprint that determines the corresponding secondary structure) in the form of RNAs is replicated and passed on to the new offspring from generation to generation. An RNA individual is then folded into its corresponding secondary structure at each generation. Fitness is then a function that measures how close is the realized structure to the target structure. Therefore, selection results when different types of RNA individuals compete with each other. One RNA may reproduce faster and out-compete the others. Occasionally, reproduction involves mistakes; these mistakes are termed mutations. Mutations are responsible for generating different RNAs that can be evaluated in the selection process, thus resulting in biological novelty and diversity.

Such a simple model gives a unified framework to precisely define and statistically measure evolutionary dynamics concepts such as plasticity, evolvability, epistasis, neutrality, continuity, and modularity. At the molecular level, plasticity is viewed as the capacity of an RNA sequence to assume a variety of energetically favourable secondary structures by equilibrating among them at a constant temperature [2]. Such concepts have been extensively studied using the RNA inverse folding as a toy model. These studies revealed that selection leads to the reduction of plasticity and, therefore, to extreme modularity [2]. Another well-studied property of evolution is neutrality which was first introduced by Kimura [97], and it suggested that the majority of genotypic changes (or mutations) in evolution are selectively neutral. The attention to Kimura's contention has led to the discovery of neutral networks in the context of genotype-phenotype models for RNA secondary structure [145, 165]. Many recent studies [178, 179] use the sequence-secondary structure relationship as a toy

model for studying neutral evolution. The neutral property of the RNA sequence-structure map contributes to a certain extent to the difficulty of the RNA design problem (e.g. when the neutral network is dense, this may quickly increase the chance of getting trapped and thus not improving the fitness). This problem is central to many optimization techniques and has already been mentioned in Chapter 4. Trying to avoid such a situation has motivated the choice of the mutation scheme implemented in `aRNAque`, which is the Lévy mutation.

Another important issue in evolutionary biology concerns the extent to which the history of life has proceeded gradually or has been punctuated by discontinuous transitions at the level of phenotypes. Distinguishing the notion of continuous from discontinuous changes at the level of phenotypes requires a notion of nearness between phenotypes. This notion was previously introduced by Fontana and Peter [57], and it is based on the probability of one phenotype being accessible from another through changes in the genotype. The RNA sequence-secondary structure relationship provides a framework where the notion of discontinuity transition is more precise. It allows understanding of how it arises in the model of evolutionary adaptation. This is done by simulating an RNA population that evolves toward a tRNA target secondary structure in a flow reactor logistically constrained to a capacity of 1000 sequences. Once the secondary target structure is found, the evolutionary trajectory is backtracked to identify all the distinct structures involved and the transition between them. An example of continuous transition in Appendix (see Figure B.2) is the transition 18 → 10 whereas the transition 15 → 22 is said to be discontinuous.

The simulation illustrated in Section B.8 was performed using `RNAfold`, the folding tool included in the `ViennaRNA` package. When using `ViennaRNA`, the plastic ensemble of an RNA sequence ϕ is often considered to be the suboptimal ensemble structure Σ_ϕ within a user-defined energy range above the MFE at a constant temperature T. The `ViennaRNA` package provides an efficient tools `RNAsubopt` allowing to compute Σ_ϕ. In a more rigorous implementation of plasticity, each of those structures in the ensemble Σ_ϕ should result from a developmental pathway. Therefore, the environmental changes may induce a change in the developmental path, allowing switching from one structure in the structural ensemble to another. When considering the set of structures produced using `RAFFT`, each meta-stable structure represents an RNA pathway; therefore, this ensemble can be considered a develop-

mental plastic ensemble. Using RAFFT to simulate the evolutionary dynamic model may provide an alternative framework to study evolutionary concepts like continuity and plasticity. Perhaps, another way of defining continuous transition $(S_1 \rightarrow S_2)$ from structure S_1 to S_2 when using will be to check if the structure S_2 is in the RAFFT's structure ensemble of the sequence with MFE S_1. In that wise, we suggest utilizing RAFFT to study and draw a different interpretation of continuous evolutionary transition.

6.4 CONCLUSION

In sum, the two computational tools introduced in Chapter 3 and Chapter 5 have been further explored. Both tools presented advantages and limitations, opening doors to further improvements and applications.

On the one hand, RAFFT predicts fast RNA pathways resulting in an ensemble of metastable structures instead of a single MFE structure implemented by most traditional methods. The ensemble structures have the advantages of containing some structures of biological relevance and reproducing complete kinetic simulations of known RNAs. However, RAFFT method presents some greediness in the choice of stems, does not provide any criterion allowing to choose biological relevant structures from the ensemble produced and does not account for barriers between structures. Despite these limitations, RAFFT offers improvements to the computational times and RNA kinetics, and its versatility opens the door to several applications from RNA design to RNA-RNA interaction.

On the other hand, aRNAque allowed designing RNA sequences with higher diversity at a reduced number of evaluations for pseudoknotted target structures. Except for CPU time performance on pseudoknotted targets, the success rate performance on both pseudoknot-free and pseudoknotted target structures showed improved performance. Despite different improvements, some Eterna100 targets remain unsolvable, opening the door to further investigations. We also discussed some aRNAque limitations, such as the influence of the pseudoknot prediction capacities of existing folding tools in the design process and aRNAque potential to control the GC-content. In addition to these two limitations, most pseudoknotted targets were solvable in less than 100 generations. These limitations contributed to the description of further research directions.

Our results go beyond the computational RNA folding and inverse folding; they can be used to study evolutionary dynamics concepts such as continuity and plasticity. Some perspectives have also been discussed.

7 GENERAL CONCLUSION

This thesis has explored computational methods for studying RNA folding. In particular, it focused on the secondary structure level. It examined the energetic and thermodynamic stability characteristics in predicting folding pathways and designing RNA target structures through inverse folding. The principal output of the thesis was the development of computational tools to efficiently predict RNA folding pathways using the FFT (`RAFFT`) and an evolutionary algorithm allowing search at both local and long-range scales in the design of target RNA structures (`aRNAque`). On the one hand, our first contribution in RNA folding, `RAFFT`, offered an alternative computational framework to predict and study the RNA kinetics for long RNA molecules at lower computation costs than classical DP methods. The versatility of our methods opened doors to different ranges of applications, such as RNA-RNA interactions and evolutionary dynamics.

On the other hand, our RNA inverse folding tool, `aRNAque`, offered a unified framework that combined the negative and positive RNA design with an EA that implements a Lévy flight mutation scheme. Our results showed general and significant improvements in the design of RNA secondary structures (especially on the pseudoknotted targets) compared to the standard evolutionary algorithm mutation scheme with a mutation parameter $\approx 1/L$, where L is the sequence solution length. Lévy flight mutation led to a greater diversity of RNA sequence solutions and reduced the evolutionary algorithm's number of evaluations, thus improving computing time compared to the local search. Although `antaRNA` average CPU time remains smaller, `aRNAque`'s success rate outperformed `antaRNA`. To further improve our program, we suggest using a more powerful computational architecture such as massively parallel genetic algorithm (MPGA). This type of architecture may allow solving more challenging target secondary structures.

Finally, we outlined these tools' limitations and prospects more generally in furthering our understanding of RNA structure, function and design. We have put them into the context of evolutionary dynamics and highlighted potential applications in studying continuous transitions and plasticity in that context. We believe

that our contributions can enhance our understanding of RNA folding and could find applications in the real world—the extensions for future works and the implications for understanding their evolution.

Part IV

APPENDIX

A

RAFFT APPENDICES

A.1 KINETIC COMPARISON

According to the RNA structure thermodynamics, one RNA molecule can adopt a structure s with probability $p(s) \propto \exp(-\beta \Delta G(s))$, where β is the inverse thermal energy (mol/kcal). To measure the quality of the ensemble of structures proposed by our method, we measured: (1) the average probability of each structures in the ensemble, then (2) the diversity of these structures.

The probability coverage PC given by $PC(s) = \frac{1}{|\Omega|} \sum_{s \in \Sigma} p(s).\, \Omega$ is the ensemble of structures sampled by a given method. We compared, for various random sequences, the probability coverage to methods based on Boltzmann sampling [46, 78]. We generated ensembles of $10^2, 10^3$, and 10^4 structures per sequence denoted respectively SB100, SB1K, and SB10K. In addition, we also compared to `RNAxplorer`, a tool also based on a biased Boltzmann sampling.

All structures are represented in the dot-bracket notation. In the dot-bracket notation, one structure has $ss = \{(, .,)\}$ symbols at each position. Given these three symbols, we propose the following positional entropy measure $S = \frac{1}{L} \sum f_i(ss) \times \log(f_i(ss))$, where $f_i(ss)$ is the frequency of a symbol ss at position i in the ensemble of structure proposed.

Figure A.1 shows the probability coverage and the positional entropy measure per method. It shows comparable sampling performances for fairly size sequences ($\approx 10^2$ nucleotides); and a comparable diversity.

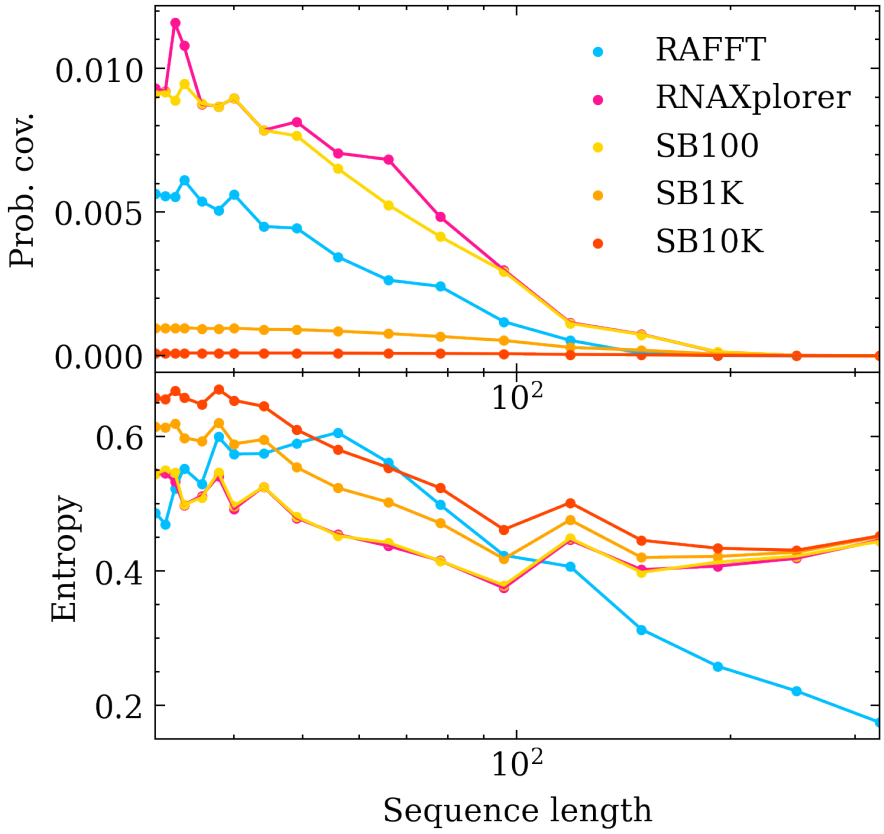


Figure A.1: **Structure ensemble characterization.** The upper part shows the average probability summed over the ensembles of structures predicted per sequence with different methods. The bottom part shows the average positional entropy of structures using the dot-bracket notation.

A.2 rafft EXAMPLE CALLS

RAFFT computes the fast-folding paths for a given sequence. Starting from the completely unfolded structure, it quickly identifies stems using the FFT-based technique.

For the examples, in the Coronavirus frameshifting stimulation element obtained from RFAM.
To display only the final structures:

Listing A.1: Command line to run RAFFT executable after installation

```
$ rafft -s
    GGGUUUGCGGUGUAAGUGCAGCCCGUCUUACACCGUGCGGCACAGGCACUAGUACUGAUGUCGUAUACAGGGCU

-ms 5
```

To display the visited/saved intermediates:

Listing A.2: Command line to run RAFFT executable after installation

```
$ rafft -s
    GGGUUUGCGGUGUAAGUGCAGCCCGUCUUACACCGUGCGGCACAGGCACUAGUACUGAUGUCGUAUACAGGGCUUUUGACAU

-ms 5
--traj
```

Here,
A result to this call could look like this:

Listing A.3: RAFFT's output results

```
GGGUUUGCGGUGUAAGUGCAGCCCGUCUUACACCGUGCGGCACAGGCACUAGUACUGAUGUCGUAUACAGGGCUUUUGACAU

# ---------0----------
.....................................................................................
        0.0
# ---------1----------
.....((((((((((.........))))))))))
    ........................................  -14.0
..................................(((((((.............))
    ))))))..............   -6.8
..................(((((.........................................)
    ))))........   -6.4
..................................((((.......))))
    ..............................   -5.5
(((((.............)))))
    ...........................................................
       -4.6
# ---------2----------
........(((((((.........)))))))((((((((((........))).))
    ))))))..............  -23.1
........(((((((.........)))))))(((((((..((........))..))
    ))))))..............  -20.9
..................(((((.........((((((((((........))).))
    ))))))...)))))........  -18.8
........(((((((.........)))))))(((((((...((....))).....))
    ))))))..............  -18.7
.....((((((((((.((.....))))))))))))
    ..................((((((.............))))))  -18.2
```

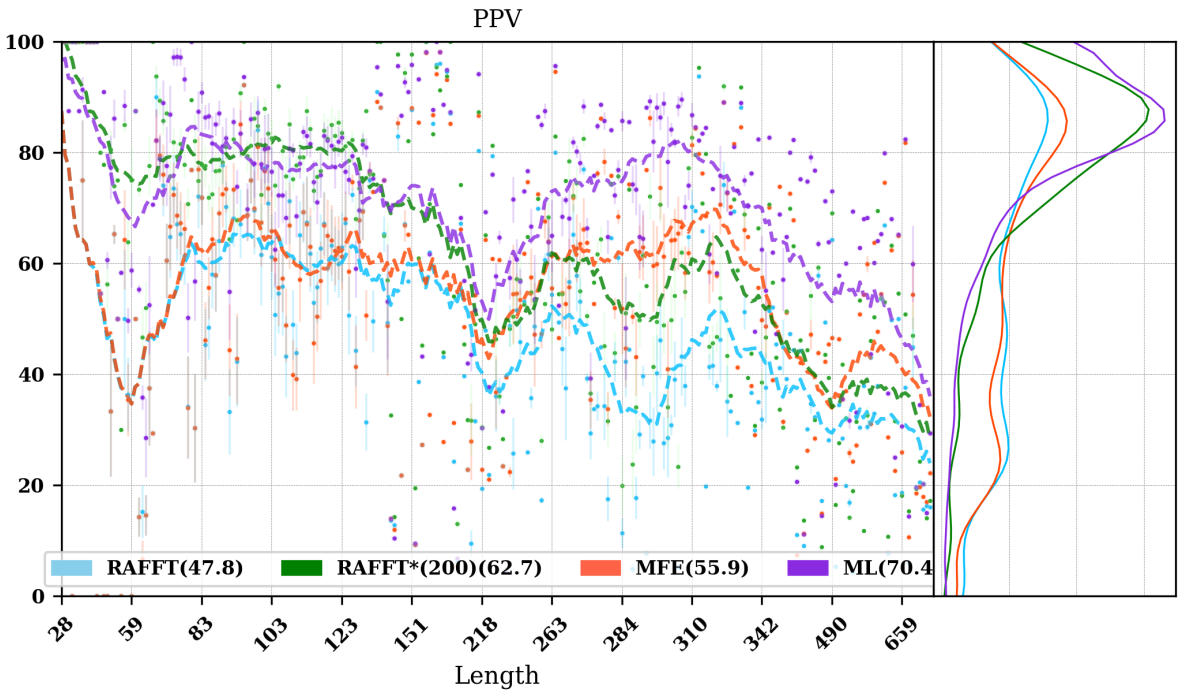
```
# ---------3----------
........(((((((.((.....)))))))))(((((((((.........))).))
   ))))))..............  -24.0
........(((((((.((.....)))))))))(((((((((...)).))).))
   ))))))..............  -24.0
........(((((((.........)))))))(((((((((.........))).))
   ))))))..............  -23.1
........(((((((.((.....)))))))))(((((((..((........))..))
   ))))))..............  -21.8
........(((((((.........)))))))(((((((..((........))..))
   ))))))..............  -20.9

GGGUUUGCGGUGUAAGUGCAGCCCGUCUUACACCGUGCGGCACAGGCACUAGUACUGAUGUCGUAUACAGGGCUUUUC

........(((((((.((.....)))))))))(((((((((.........))).))
   ))))))..............  -24.0
........(((((((.((.....)))))))))(((((((((...)).))).))
   ))))))..............  -24.0
........(((((((.........)))))))(((((((((.........))).))
   ))))))..............  -23.1
........(((((((.((.....)))))))))(((((((..((........))..))
   ))))))..............  -21.8
........(((((((.........)))))))(((((((..((........))..))
   ))))))..............  -20.9
```

Where the columns shows respectively the predicted structures and their free energy.

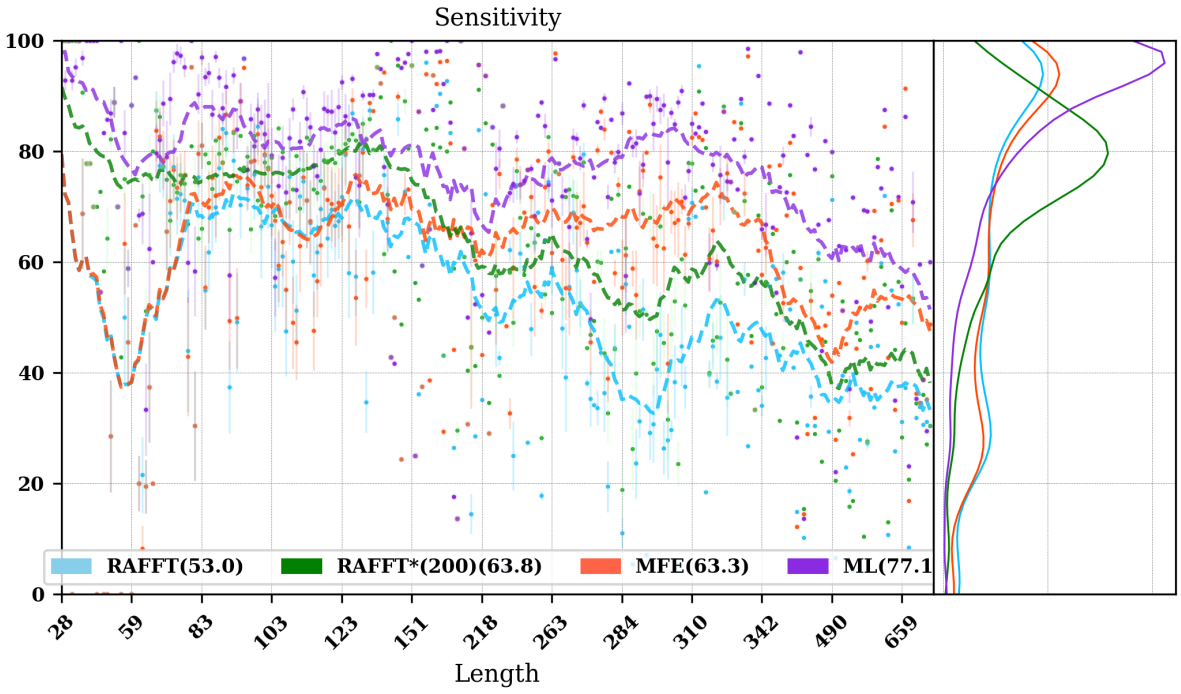
A.3 rafft performance analysis for 200 structures saved.

A.4 rafft performance analysis with various values of minimum energy contribution required for loop formation

A.5

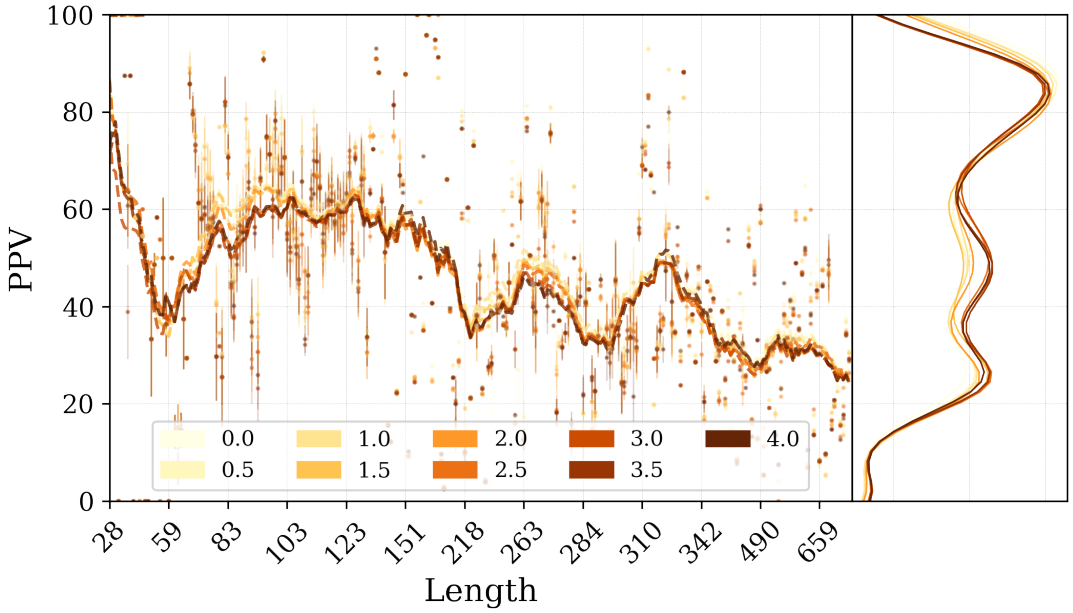


(a)

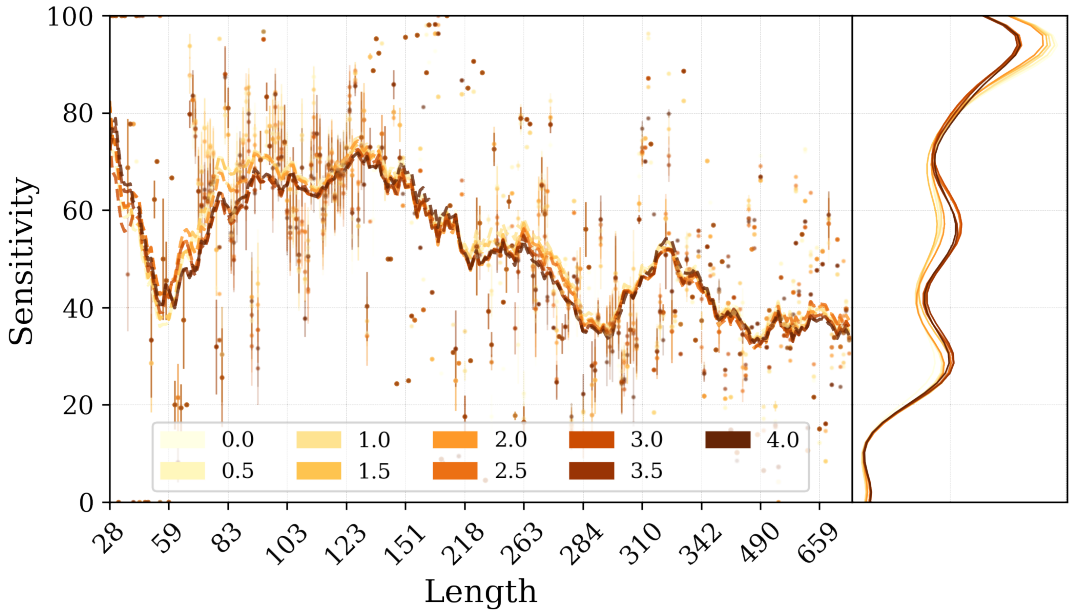


(b)

Figure A.2: **Positive predictive values and sensitivity results.** RAFFT (blue) displayed the best energy found. RAFFT*(200) shows the best score found among 200 saved structures. Left pans show the density (sequence-wise) of the accuracy measures.



(a)



(b)

Figure A.3: **Predictive performance of RAFFT with various values of minimum energy contribution required for loop formation.** Positive values for this parameter causes RAFFT to accept destabilizing loops, therefore being less greedy than per default. The performance of RAFFT was not observed to be positively affected by allowing sub-optimal loop formation.

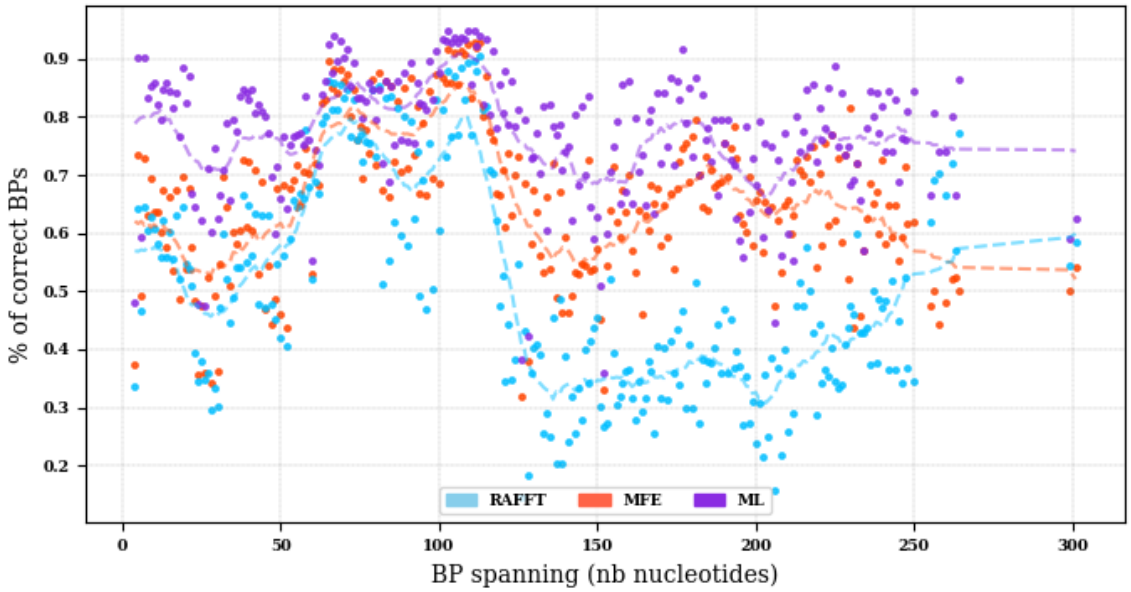


Figure A.4: **Base pair spanning: It shows the percent of base pairs predicted found in the known structures per number of nucleotides between them.**

B

ARNAQUE APPENDICES

B.1 arnaque's gc-content parameters

The GC-content is controlled in aRNAque using the mutation parameters P_C and P_N. The following table gives the corresponding mutation parameters to the four regimes of GC-content values used for our benchmark.

Table B.1: **Mutation parameters used in aRNAque to control the GC–content values**.

GC–content values	P_C	P_N	aRNAque's key
0.25	$\{0.125, 0.125, 0.3, 0.3, 0.075, 0.075\}$	$\{0.125, 0.125, 0.375, 0.375\}$	GC_{25}
0.25	$\{0.25, 0.25, 0.2, 0.2, 0.05, 0.05\}$	$\{0.25, 0.25, 0.25, 0.5\}$	GC_{50}
0.75	$\{0.375, 0.375, 0.1, 0.1, 0.025, 0.025\}$	$\{0.375, 0.375, 0.125, 0.125\}$	GC_{75}
1.0	$\{0.5, 0.5, 0.0, 0.0, 0.0, 0.0\}$	$\{0.5, 0.5, 0., 0.\}$	GC

B.2 BENCHMARK ON eterna100 DATASET

For each of the benchmarks on the Eterna100 datasets, We ran the first benchmark using the default aRNAque's parameter configuration. And then, the unsolved structures are sorted out to run a second benchmark with a maximum number of generations set at 5000. aRNAque's performance presented in the paper is a combination of all the designed sequences for each realisation.

B.3 GENERAL EA BENCHMARK PARAMETERS

The same hardware resources and the same computer are used for all the benchmarks listed in the following table. A supercomputer with 40-Core Intel Xeon E5-2698 v4 at 2.2 GHz and 512 GB of RAM with a Debian OS.

Table B.2: **Evolutionary algorithm parameter for each benchmarks**.

Benchmark	Population size	# of generations (T)	Stopping criterion	Mutation parameter	# of runs per target
PseudoBase++ (`IPknot`)	100	200	$t = T$ $\max(f) = 0$	$c = 1.5$ $P_N = \{0.7, 0.1, 0.1, .1\}$ $P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$	20
PseudoBase++ (`Hotknots`)	100	200	$t = T$ $\max(f) = 0$	$c = 1.5$ $P_N = \{0.7, 0.1, 0.1, .1\}$ $P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$	20
PseudoBase++ GC–content (`IPknot`)	100	200	$t = T$ $\max(f) = 0$	$c = 1.5$ $P_N = \{0.7, 0.1, 0.1, .1\}$ $P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$	20
Tuning Parameter (`Binomial`,`IPknot`)	100	200	$t = T$ $\max(f) = 0$	$\mu \in [0, 0.2]; c = None$ $P_N = \{0.7, 0.1, 0.1, .1\}$ $P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$	20
Tuning Parameter (`Lévy`, `IPknot`)	100	200	$t = T$ $\max(f) = 0$	$c \in [1, 2]; \mu = None$ $P_N = \{0.7, 0.1, 0.1, .1\}$ $P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$	20
`Eterna100-V1` (OP, `RNAfold`)	100	5000	$t = T$ $\max(f) = 0$	$c = 7$ $P_N = \{0.7, 0.1, 0.1, .1\}$ $P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$	5
`Eterna100-V1` (Lévy, `RNAfold`)	100	5000	$t = T$ $\max(f) = 0$	$c = 1$ $P_N = \{0.7, 0.1, 0.1, 0.1\}$ $P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$	5
`Eterna100-V2` (OP, `RNAfold`)	100	5000	$t = T$ $\max(f) = 0$	$c = 7$ $P_N = \{0.7, 0.1, 0.1, .1\}$ $P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$	5
`Eterna100-V2` (Lévy, `RNAfold`)	100	5000	$t = T$ $\max(f) = 0$	$c = 1.5$ $P_N = \{0.7, 0.1, 0.1, .1\}$ $P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$	5

B.4 OTHER BENCHMARK ON eterna100-v1

The results on Eterna100-V1 presented in the paper are the best of all the benchmarks we have performed. Since our mutation scheme relies on the nucleotide distributions which implicitly control the GC–content of the designed sequences, to obtain our results, we first selected an arbitrary set of pairs $\{P_N, P_C\}$ and benchmark aRNAque on Eterna100-V1 for each of them. The success rate measures the fraction of sequences successfully folding into the target structure. Table **??** shows the different parameters we considered and the corresponded input key parameter using the call of aRNAque script. Summary of the benchmark presented in Table **??** is obtained by launching for each target structure 5 independent runs, with a population size of 100 and a maximum number of generations of 5000. The energy parameter used here was the Turner1999. The dashes in the table mean the benchmarks have not been performed for the parameters.

Table B.3: **Different parameters for the base pair distributions**

Key	$P_N = \{p_A, p_G, p_U, p_C\}$	$P_C = \{p_{GC}, p_{CG}, p_{AU}, p_{UA}, p_{GU}, p_{UG}\}$
ALL	$P_N = \{0.25, 0.25, 0.25, 0.25\}$	$P_C = \{0.2, 0.2, 0.1, 0.1, 0.2, 0.2\}$
GC	$P_N = \{0.25, 0.25, 0.25, 0.25\}$	$P_C = \{0.5, 0.5, 0, 0, 0, 0\}$
GC_1	$P_N = \{0.25, 0.65, 0.05, 0.05\}$	$P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$
GC_2	$P_N = \{0.7, 0.1, 0.1, 0.1\}$	$P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$
GC_3	$P_N = \{0.75, 0.1, 0.1, 0.05\}$	$P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$
GC_4	$P_N = \{0.95, 0, 0.05, 0\}$	$P_C = \{0.4, 0.4, 0.2, 0, 0, 0\}$
GC_5	$P_N = \{0.7, 0.1, 0.1, 0.1\}$	$P_C = \{0.3, 0.2, 0.2, 0.1, 0.1, 0.1\}$

Table B.4: **Success percentage on `Eterna100` datasets for each set of mutation parameters.**

Tools	BP param	Mutation param	Percentage of success	$\#(Med(gen_{Zipf}) < Med(gen_{op}))$ $\#(Med(gen_{Zipf}) > Med(gen_{op}))$
aRNAque	*ALL*	Zipf ($c = 1$) One point	67% 81%	7(#4) 64(#4)
aRNAque	*GC*	Zipf ($c = 1$) One point	80% 90%	43(#10) 30(#474)
aRNAque	GC_1	Zipf ($c = 1$) One point	84% 90%	29(#4) 33(#7)
aRNAque	GC_2	Zipf ($c = 1$) One point	89% 91%	61(#10) 19(#1920)
aRNAque	GC_3	Zipf ($c = 1$) One point	88% −−	−− −−
aRNAque	GC_4	Zipf ($c = 1$) One point	−− −−	−− −−
aRNAque	GC_5	Zipf ($c = 1$) One point	82% 83%	44(#9) 30(#145)
Total	–	Zipf ($c = 1$) One point RNAinverse	**90%** **92%** **87%**	

B.5 TOOLS PATCHING

To be able to perform our benchmarks, some slight modifications was made on `HotKnots` and `antaRNA`. Details about the modifications are provided in this section.

- `antaRNA`: The change was made at the line 1178 column 7, where the line args = 'HotKnots -m CC -s ' + sequence was replaced by to args = './HotKnots -m CC -s ' + sequence. The version of `antaRNA` we used is v2.0.1, and it can be found on the Github link: `https://github.com/RobertKleinkauf/antarna`.

- `HotKnots`: to run `HotKnots`, we have to move `aRNAque` to the bin directory. To avoid that, we updated the source code and recompiled a new bin that does not require to move `aRNAque` to the bin directory of `HotKnots`. We have uploaded

the patched version of HotKnots in a third-part folder in aRNAque's repository for benchmark reproduction.

NB: The patches do not affect the folding algorithm. It consisted of avoiding the use of relative paths in HotKnots.

B.6 arnaque EXAMPLE CALLS

aRNAque computes the RNA inverse folding problem for different classes of structure complexities.
For a pseudo-knot free target secondary structure:

Listing B.1: Command line to run aRNAque python script

```
$ python aRNAque.py -t "((....)).((....))"
                        -bp "GC2"
                        -sm "NHD"
                        -ft "v"
                        --job 5
```

Here,
A result to this call could look like this:

Listing B.2: aRNAque's output results

```
GCUACGGCACCGUCAGG | ((....)).((....)) | -2.8 | 1.0
GGGGGACCACCGGUGGG | ((....)).((....)) | -2.5 | 1.0
GGGCCACCAGCGAAAGC | ((....)).((....)) | -2.2 | 1.0
GGAAAUCCACCGGAAGG | ((....)).((....)) | -1.4 | 1.0
GCAAGAGCGCCGCAAGG | ((....)).((....)) | -1.2 | 1.0
```

Where the columns shows respectively the designed sequences, the MFE structures, their free energy and the fitness to the target (See Equation 5.1)

B.7 LÉVY FLIGHT VS LOCAL SEARCH: DESIGNING THE STRUCTURE WITH THE SMALLEST NEUTRAL SET IN THE SPACE OF ALL RNA SEQUENCES OF LENGTH 12

To further illustrate that advantage, we considered the space of all RNA sequences of length 12 and with only G,C nucleotides. The structures with the lowest neutral set are:

1. $T_1 = ((((...)).))$: only 2 sequences fold into the secondary structure T_1
2. $T_2 = ((.((...))))$: only 1 sequence folds into the secondary structure T_2

When having a close look at those two structures the base pair density is maximal and there is an unpaired position on both that allows the formation of a budge.

What that means naively is that any compatible sequence to T_1 (or T_2) will likely fold into a stem with four or three base pairs(((((...)))). Or (((....)))..) , and these particular structures have respectively 243 and 249 sequences in their neutral sets.

We claim that, when having such kind of structure (T_1 or T_2), the levy mutation is of an important role to get out of the huge neutral network of more stable stems. A simple test case was to run `aRNAque` for a target secondary structure T_1. For both one point and Lévy mutations, the distribution of the number of generations needed to find sequences that fold into T_1 for both mutation schemes is plotted in Figure B.1.

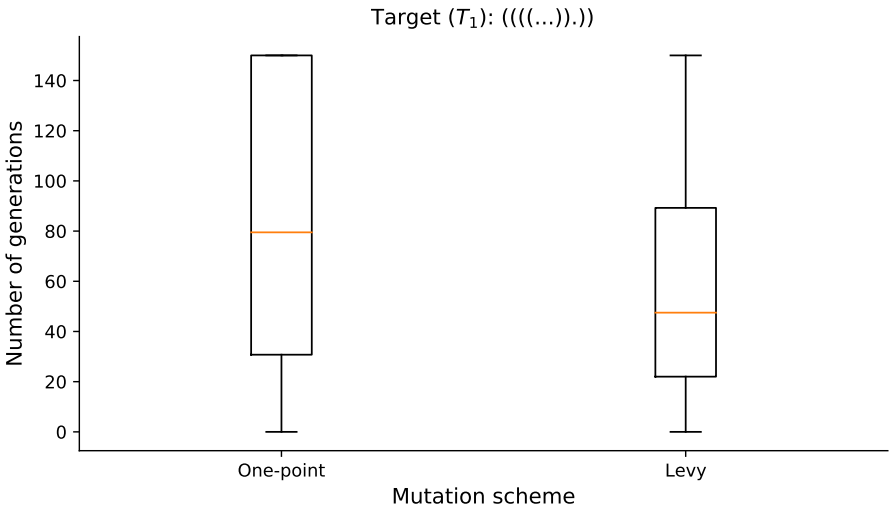
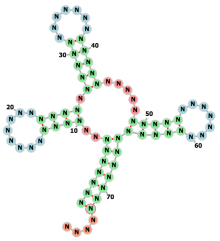


Figure B.1: **Distribution of number of generations need to solve the target T_1, for both Lévy and Local mutation schemes.**

B.8 CONTINUOUS AND DISCONTINUOUS TRANSITIONS IN EVOLUTION

Figure B.2 shows the evolution of the average distance to the tRNA target structure, the intervals of time for which a particular structure is present in the population, and a transition between

distinct structures present in the evolutionary path. In Fontana's suggestions, a transition $(S_1 \rightarrow S_2)$ between two structures S_1 and S_2 is considered to be continuous if the structure S_1 is 'near' S_2. In other terms, S_2 is likely to be accessible through the neighbour neutral sequences of S_1. So if S_2 appears in the evolutionary path at time t, there exists a time $t' < t$ where S_2 was already present in the population. In contrast, the transition is discontinuous otherwise (i.e. the time the structure S_2 appears in the evolutionary path exactly at the same time it was present in the population). An example of continuous transition in Figure B.2 is the transition $18 \rightarrow 10$ whereas the transition $15 \rightarrow 22$ is said to be discontinuous.



tRNA target secondary structure..

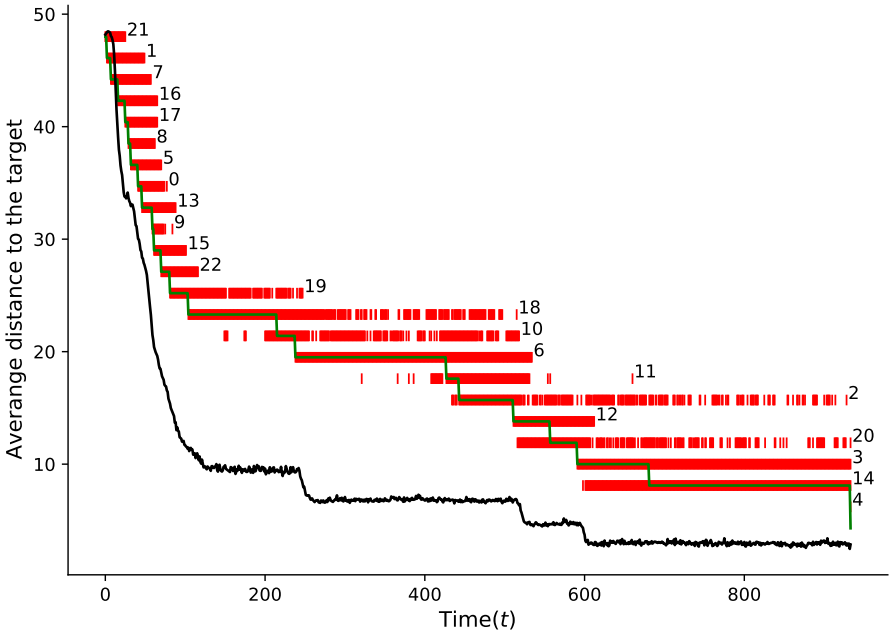


Figure B.2: **Simulation of an RNA population evolving toward a tRNA (See the figure on the right side) target secondary structure**. The target was reached after 933 generations (i.e. $\approx 10^5$ replications). The black line shows the average structure distance of the structures in the population to the target structure. The evolutionary history linking the initial structure to the target structure comprises 23. Each structure is labelled by an integer taken from 0 to 22. To each of them corresponds one horizontal line (in red). The top-level corresponds to the initial structure and the bottom the target structure. At each level, a series of red intervals correspond to the periods when the structure was present in the population, and the green curve represents the transition between structures. Only the time axis has a meaning for the red and green curves.

BIBLIOGRAPHY

[1] Fabian Amman, Stephan H. Bernhart, Gero Doose, Ivo L. Hofacker, Jing Qin, Peter F. Stadler, and Sebastian Will. "The trouble with long-range base pairs in RNA folding." In: *Advances in Bioinformatics and Computational Biology*. Advances in Bioinformatics and Computational Biology. Springer International Publishing, 2013, pp. 1–11. DOI: 10.1007/978-3-319-02624-4_1. URL: https://doi.org/10.1007/978-3-319-02624-4_1.

[2] Lauren W Ancel and Walter Fontana. "Plasticity, evolvability, and modularity in RNA." In: *Journal of Experimental Zoology* 288.3 (2000), pp. 242–283.

[3] Jeff Anderson-Lee, Eli Fisker, Vineet Kosaraju, Michelle Wu, Justin Kong, Jeehyung Lee, Minjae Lee, Mathew Zada, Adrien Treuille, and Rhiju Das. "Principles for predicting RNA secondary structure design difficulty." In: *Journal of molecular biology* 428.5 (2016), pp. 748–757.

[4] Mirela Andronescu, Vera Bereg, Holger H Hoos, and Anne Condon. "RNA STRAND: the RNA secondary structure and statistical analysis database." In: *BMC Bioinformatics* 9.1 (2008), pp. 1–10.

[5] Mirela Andronescu, Anthony P Fejes, Frank Hutter, Holger H Hoos, and Anne Condon. "A new algorithm for RNA secondary structure design." In: *Journal of molecular biology* 336.3 (2004), pp. 607–624.

[6] Assaf Avihoo, Alexander Churkin, and Danny Barash. "RNAexinv: An extended inverse RNA folding from shape and physical attributes to sequences." In: *BMC bioinformatics* 12.1 (2011), pp. 1–8.

[7] Rolf Backofen and Wolfgang R Hess. "Computational prediction of sRNAs and their targets in bacteria." In: *RNA biology* 7.1 (2010), pp. 33–42.

[8] Rodolphe Barrangou, Christophe Fremaux, Hélène Deveau, Melissa Richards, Patrick Boyaval, Sylvain Moineau, Dennis A Romero, and Philippe Horvath. "CRISPR provides acquired resistance against viruses in prokaryotes." In: *Science* 315.5819 (2007), pp. 1709–1712.

[9] S. Bellaousov and D. H. Mathews. "Probknot: fast prediction of RNA secondary structure including pseudoknots." In: *RNA* 16.10 (2010), pp. 1870–1880. DOI: 10.1261/rna.2125310. URL: https://doi.org/10.1261/rna.2125310.

[10] Richard Bellman. "Dynamic programming, princeton univ." In: *Press Princeton, New Jersey* (1957).

[11] Jan H Bergmann and David L Spector. "Long non-coding RNAs: modulators of nuclear structure and function." In: *Current opinion in cell biology* 26 (2014), pp. 10–18.

[12] Philip C Bevilacqua, Laura E Ritchey, Zhao Su, and Sarah M Assmann. "Genome-wide analysis of RNA secondary structure." In: *Annual review of genetics* 50 (2016), pp. 235–266.

[13] Eckart Bindewald, Kirill Afonin, Luc Jaeger, and Bruce A Shapiro. "Multistrand RNA secondary structure prediction and nanostructure design including pseudoknots." In: *ACS nano* 5.12 (2011), pp. 9542–9551.

[14] Édouard Bonnet, Paweł Rzążewski, and Florian Sikora. "Designing RNA secondary structures is hard." In: *Journal of Computational Biology* 27.3 (2020), pp. 302–316.

[15] Ronald R Breaker, RF Gesteland, TR Cech, and JF Atkins. *The RNA world*. Cold Spring Harbor Laboratory Press, New York, 2006.

[16] Philippe Brion and Eric Westhof. "Hierarchy and dynamics of RNA folding." In: *Annual review of biophysics and biomolecular structure* 26.1 (1997), pp. 113–137.

[17] James W Brown. "The ribonuclease P database." In: *Nucleic Acids Research* 26.1 (1998), pp. 351–352.

[18] Anke Busch and Rolf Backofen. "INFO-RNA—a fast approach to inverse RNA folding." In: *Bioinformatics* 22.15 (2006), pp. 1823–1831.

[19] Thomas R Cech and Joan A Steitz. "The noncoding RNA revolution—trashing old rules to forge new ones." In: *Cell* 157.1 (2014), pp. 77–94.

[20] Shaon Chakrabarti, Changbong Hyeon, Xiang Ye, George H Lorimer, and D Thirumalai. "Molecular chaperones maximize the native state yield on biological times by driving substrates out of equilibrium." In: *Proceedings of the National Academy of Sciences* 114.51 (2017), E10919–E10927.

[21] James Chappell, Kyle E Watters, Melissa K Takahashi, and Julius B Lucks. "A renaissance in RNA synthetic biology: new mechanisms, applications and tools for the future." In: *Current opinion in chemical biology* 28 (2015), pp. 47–56.

[22] Shi-Jie Chen. "RNA folding: conformational statistics, folding kinetics, and ion electrostatics." In: *Annu. Rev. Biophys.* 37 (2008), pp. 197–214.

[23] Alexander Churkin, Matan Drory Retwitzer, Vladimir Reinharz, Yann Ponty, Jérôme Waldispühl, and Danny Barash. "Design of RNAs: comparing programs for inverse RNA folding." In: *Briefings in bioinformatics* 19.2 (2017), pp. 350–358.

[24] Simona Cocco, John F Marko, and Remi Monasson. "Slow nucleic acid unzipping kinetics from sequence-defined barriers." In: *The European Physical Journal E* 10.2 (2003), pp. 153–161.

[25] James W Cooley and John W Tukey. "An algorithm for the machine calculation of complex Fourier series." In: *Mathematics of computation* 19.90 (1965), pp. 297–301.

[26] Francis Crick. "Central dogma of molecular biology." In: *Nature* 227.5258 (1970), pp. 561–563.

[27] Emilio Cusanelli and Pascal Chartrand. "Telomeric noncoding RNA: telomeric repeat-containing RNA in telomere biology." In: *Wiley Interdisciplinary Reviews: RNA* 5.3 (2014), pp. 407–419.

[28] Paul Dallaire and François Major. "Exploring alternative RNA structure sets using MC-flashfold and db2cm." In: *RNA Structure Determination*. Springer, 2016, pp. 237–251.

[29] Simon H Damberger and Robin R Gutell. "A comparative database of group I intron structures." In: *Nucleic Acids Research* 22.17 (1994), pp. 3508–3510.

[30] Christian Darabos, Mario Giacobini, Ting Hu, and Jason H. Moore. "Lévy-Flight Genetic Programming: Towards a New Mutation Paradigm." In: *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Ed. by Mario Giacobini, Leonardo Vanneschi, and William S. Bush. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 38–49. ISBN: 978-3-642-29066-4.

[31] Kévin Darty, Alain Denise, and Yann Ponty. "VARNA: Interactive drawing and editing of the RNA secondary structure." In: *Bioinformatics* 25.15 (2009), p. 1974.

[32] Jennifer Daub, Paul P Gardner, John Tate, Daniel Ramsköld, Magnus Manske, William G Scott, Zasha Weinberg, Sam Griffiths-Jones, and Alex Bateman. "The RNA WikiProject: community annotation of RNA families." In: *RNA* 14.12 (2008), pp. 2462–2464.

[33] Christoph Dieterich and Peter F Stadler. "Computational biology of RNA interactions." In: *Wiley Interdisciplinary Reviews: RNA* 4.1 (2013), pp. 107–120.

[34] Ken A Dill. "Additivity principles in biochemistry." In: *Journal of Biological Chemistry* 272.2 (1997), pp. 701–704.

[35] Robert M Dirks, Milo Lin, Erik Winfree, and Niles A Pierce. "Paradigms for computational nucleic acid design." In: *Nucleic acids research* 32.4 (2004), pp. 1392–1403.

[36] Robert M. Dirks and Niles A. Pierce. "A partition function algorithm for nucleic acid secondary structure including pseudoknots." In: *Journal of Computational Chemistry* 24.13 (2003). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.10296, pp. 1664–1677. ISSN: 1096-987X. DOI: https://doi.org/10.1002/jcc.10296. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.10296 (visited on 02/17/2021).

[37] Robert M Dirks and Niles A Pierce. "A partition function algorithm for nucleic acid secondary structure including pseudoknots." In: *Journal of computational chemistry* 24.13 (2003), pp. 1664–1677.

[38] Chuong B Do, Daniel A Woods, and Serafim Batzoglou. "CONTRAfold: RNA secondary structure prediction without physics-based models." In: *Bioinformatics* 22.14 (2006), e90–e98.

[39] Elizabeth A Doherty and Jennifer A Doudna. "Ribozyme structures and mechanisms." In: *Annual Review of Biophysics and Biomolecular Structure* 30.1 (2001), pp. 457–475.

[40] Ivan Dotu, Juan Antonio Garcia-Martin, Betty L Slinger, Vinodh Mechery, Michelle M Meyer, and Peter Clote. "Complete RNA inverse folding: computational design of functional hammerhead ribozymes." In: *Nucleic acids research* 42.18 (2014), pp. 11752–11762.

[41] Robin D Dowell and Sean R Eddy. "Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction." In: *BMC bioinformatics* 5.1 (2004), pp. 1–14.

[42] D Draper, T Gluick, and P Schlax. *In RNA Structure and Function, Vol. 298*. 2000.

[43] N Dromi, A Avihoo, and D Barash. "Reconstruction of natural RNA sequences from RNA shape, thermodynamic stability, mutational robustness, and linguistic complexity by evolutionary computation." In: *Journal of Biomolecular Structure and Dynamics* 26.1 (2008), pp. 147–161.

[44] Matan Drory Retwitzer, Vladimir Reinharz, Yann Ponty, Jérôme Waldispühl, and Danny Barash. "incaRNAfbinv: a web server for the fragment-based design of RNA sequences." In: *Nucleic acids research* 44.W1 (2016), W308–W314.

[45] Andrew D Ellington, Xi Chen, Michael Robertson, and Angel Syrett. "Evolutionary origins and directed evolution of RNA." In: *The international journal of biochemistry & cell biology* 41.2 (2009), pp. 254–265.

[46] Gregor Entzian, Ivo L Hofacker, Yann Ponty, Ronny Lorenz, and Andrea Tanzer. "RNAxplorer: harnessing the power of guiding potentials to sample RNA landscapes." In: *Bioinformatics* 37.15 (2021), pp. 2126–2133.

[47] Ali Esmaili-Taheri and Mohammad Ganjtabesh. "ERD: a fast and reliable tool for RNA design including constraints." In: *BMC bioinformatics* 16.1 (2015), p. 20.

[48] Ali Esmaili-Taheri, Mohammad Ganjtabesh, and Morteza Mohammad-Noori. "Evolutionary solution for the RNA design problem." In: *Bioinformatics* 30.9 (2014), pp. 1250–1258.

[49] Manel Esteller. "Non-coding RNAs in human disease." In: *Nature reviews genetics* 12.12 (2011), pp. 861–874.

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

[51] Alessandro Fatica and Irene Bozzoni. "Long non-coding RNAs: new players in cell differentiation and development." In: *Nature Reviews Genetics* 15.1 (2014), pp. 7–21.

[52] Sven Findeiß, Manja Wachsmuth, Mario Mörl, and Peter F Stadler. "Design of transcription regulating riboswitches." In: *Methods in enzymology*. Vol. 550. Elsevier, 2015, pp. 1–22.

[53] Christoph Flamm, Walter Fontana, Ivo L Hofacker, and Peter Schuster. "RNA folding at elementary step resolution." In: *Rna* 6.3 (2000), pp. 325–338.

[54] Christoph Flamm, Ivo L Hofacker, Sebastian Maurer-Stroh, Peter F Stadler, and Martin Zehl. "Design of multistable RNA molecules." In: *Rna* 7.2 (2001), pp. 254–265.

[55] Christoph Flamm, Ivo L. Hofacker, Peter F. Stadler, and Michael T. Wolfinger. "Barrier trees of degenerate landscapes." In: *Zeitschrift für Physikalische Chemie* 216.2 (2002), nil. DOI: 10.1524/zpch.2002.216.2.155. URL: https://doi.org/10.1524/zpch.2002.216.2.155.

[56] Peter J Flor, James B Flanegan, and TR Cech. "A conserved base pair within helix P4 of the Tetrahymena ribozyme helps to form the tertiary structure required for self-splicing." In: *The EMBO Journal* 8.11 (1989), pp. 3391–3399.

[57] Walter Fontana and Peter Schuster. "Continuity in evolution: on the nature of transitions." In: *Science* 280.5368 (1998), pp. 1451–1455.

[58] Jacques R. Fresco, Bruce M. Alberts, and Paul Doty. "Some Molecular Details of the Secondary Structure of Ribonucleic Acid." In: *Nature* 188.4745 (Oct. 1960). Number: 4745 Publisher: Nature Publishing Group, pp. 98–101. ISSN: 1476-4687. DOI: 10.1038/188098a0. URL: https://www.nature.com/articles/188098a0 (visited on 04/14/2021).

[59] James ZM Gao, Linda YM Li, and Christian M Reidys. "Inverse folding of RNA pseudoknot structures." In: *Algorithms for Molecular Biology* 5.1 (2010), pp. 1–19.

[60] Juan Antonio Garcia-Martin, Peter Clote, and Ivan Dotu. "RNAiFOLD: a constraint programming algorithm for RNA inverse folding and molecular design." In: *Journal of bioinformatics and computational biology* 11.02 (2013), p. 1350001.

[61] Paul P Gardner, Jennifer Daub, John G Tate, Eric P Nawrocki, Diana L Kolbe, Stinus Lindgreen, Adam C Wilkinson, Robert D Finn, Sam Griffiths-Jones, Sean R Eddy, et al. "Rfam: updates to the RNA families database." In: *Nucleic Acids Research* 37.suppl_1 (2009), pp. D136–D140.

[62] Michael Geis, Christoph Flamm, Michael T Wolfinger, Andrea Tanzer, Ivo L Hofacker, Martin Middendorf, Christian Mandl, Peter F Stadler, and Caroline Thurner. "Folding kinetics of large RNAs." In: *Journal of Molecular Biology* 379.1 (2008), pp. 160–173.

[63] David P Giedroc, Carla A Theimer, and Paul L Nixon. "Structure, stability and function of RNA pseudoknots involved in stimulating ribosomal frameshifting." In: *Journal of molecular biology* 298.2 (2000), pp. 167–185.

[64] Sunny D Gilbert, Robert P Rambo, Daria Van Tyne, and Robert T Batey. "Structure of the SAM-II riboswitch bound to S-adenosylmethionine." In: *Nature structural & molecular biology* 15.2 (2008), pp. 177–182.

[65] Walter Gilbert. "Origin of life: The RNA world." In: *Nature* 319.6055 (Feb. 1986). Number: 6055 Publisher: Nature Publishing Group, pp. 618–618. ISSN: 1476-4687. DOI: 10.1038/319618a0. URL: https://www.nature.com/articles/319618a0 (visited on 04/09/2021).

[66] Sam F Greenbury, Steffen Schaper, Sebastian E Ahnert, and Ard A Louis. "Genetic correlations greatly increase mutational robustness and can both reduce and enhance evolvability." In: *PLoS computational biology* 12.3 (2016), e1004773.

[67] A. P. Gultyaev, F. H. van Batenburg, and C. W. Pleij. "An approximation of loop free energy values of RNA H-pseudoknots." In: *RNA (New York, N.Y.)* 5.5 (May 1999), pp. 609–617. ISSN: 1355-8382. DOI: 10.1017/s135583829998189x.

[68] Peixuan Guo. "The emerging field of RNA nanotechnology." In: *Nature nanotechnology* 5.12 (2010), pp. 833–842.

[69] Zhuyan Guo and D Thirumalai. "Kinetics of protein folding: nucleation mechanism, time scales, and pathways." In: *Biopolymers: Original Research on Biomolecules* 36.1 (1995), pp. 83–102.

[70] Robin R Gutell. "Collection of small subunit (16S-and 16S-like) ribosomal RNA structures: 1994." In: *Nucleic Acids Research* 22.17 (1994), pp. 3502–3507.

[71] Robin R Gutell, Michael W Gray, and Murray N Schnare. "A compilation of large subunit (23S and 23S-like) ribosomal RNA structures: 1993." In: *Nucleic Acids Research* 21.13 (1993), p. 3055.

[72] Robin R Gutell, Jung C Lee, and Jamie J Cannone. "The accuracy of ribosomal RNA comparative structure models." In: *Current opinion in structural biology* 12.3 (2002), pp. 301–310.

[73] Robin R Gutell, Bryn Weiser, Carl R Woese, and Harry F Noller. "Comparative anatomy of 16-S-like ribosomal RNA." In: *Progress in nucleic acid research and molecular biology* 32 (1985), pp. 155–216.

[74] Christine E Hajdin, Stanislav Bellaousov, Wayne Huggins, Christopher W Leonard, David H Mathews, and Kevin M Weeks. "Accurate SHAPE-directed RNA secondary structure modeling, including pseudoknots." In: *Proceedings of the National Academy of Sciences* 110.14 (2013), pp. 5498–5503.

[75] Arif Ozgun Harmanci, Gaurav Sharma, and David H Mathews. "Stochastic sampling of the RNA structural alignment space." In: *Nucleic acids research* 37.12 (2009), pp. 4063–4075.

[76] Christian Haslinger and Peter F. Stadler. "RNA structures with pseudo-knots: Graph-theoretical, combinatorial, and statistical properties." In: *Bulletin of Mathematical Biology* 61.3 (May 1, 1999), pp. 437–467. ISSN: 1522-9602. DOI: 10.1006/bulm.1998.0085. URL: https://doi.org/10.1006/bulm.1998.0085 (visited on 04/15/2021).

[77] Teresa Haynes, Debra Knisley, and Jeff Knisley. "Using a neural network to identify secondary RNA structures quantified by graphical invariants." In: *Comm Math Comput Chem* 60 (2008), pp. 277–290.

[78] I. L. Hofacker. "Vienna RNA secondary structure server." In: *Nucleic Acids Research* 31.13 (2003), pp. 3429–3431. DOI: 10.1093/nar/gkg599. URL: https://doi.org/10.1093/nar/gkg599.

[79] Ivo L. Hofacker. "RNA Secondary Structure Prediction." In: *eLS*. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1038/npg.els.0005274. American Cancer Society, 2005. ISBN: 978-0-470-01590-2. DOI: 10.1038/npg.els.0005274. URL: https://onlinelibrary.wiley.com/doi/abs/10.1038/npg.els.0005274 (visited on 03/03/2021).

[80] Ivo L Hofacker, Walter Fontana, Peter F Stadler, L Sebastian Bonhoeffer, Manfred Tacker, and Peter Schuster. "Fast folding and comparison of RNA secondary structures." In: *Monatshefte für Chemie/Chemical Monthly* 125.2 (1994), pp. 167–188.

[81] Ivo L. Hofacker, Peter F. Stadler, and Peter F. Stadler. "RNA Secondary Structures." In: *Reviews in Cell Biology and Molecular Medicine*. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/3527600906.mcb.200500009. American Cancer Society, 2006. ISBN: 978-3-527-60090-8. DOI: 10.1002/3527600906.mcb.200500009. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/3527600906.mcb.200500009 (visited on 03/03/2021).

[82] J Holland. *Adaptation in Natural and Artificial Systems, Ann Arbor: University of Michigan Press, 1975*. 1992.

[83] Chiou-Yi Hor, Chang-Biau Yang, Chia-Hung Chang, Chiou-Ting Tseng, and Hung-Hsin Chen. "A Tool preference choice Method for RnA secondary structure prediction by sVM with statistical Tests." In: *Evolutionary Bioinformatics* 9 (2013), EBO–S10580.

[84] Liang Huang, He Zhang, Dezhong Deng, Kai Zhao, Kaibo Liu, David A Hendrix, and David H Mathews. "LinearFold: linear-time approximate RNA folding by 5'-to-3'dynamic programming and beam search." In: *Bioinformatics* 35.14 (2019), pp. i295–i304.

[85] Woong Y Hwang, Yanfang Fu, Deepak Reyon, Morgan L Maeder, Shengdar Q Tsai, Jeffry D Sander, Randall T Peterson, J-R Joanna Yeh, and J Keith Joung. "Efficient in vivo genome editing using RNA-guided nucleases." In: *Nature biotechnology* 31.3 (2013), p. 227.

[86] Farren J Isaacs, Daniel J Dwyer, and James J Collins. "RNA synthetic biology." In: *Nature biotechnology* 24.5 (2006), pp. 545–554.

[87] Hervé Isambert and Eric D Siggia. "Modeling RNA folding paths with pseudoknots: application to hepatitis delta virus ribozyme." In: *Proceedings of the National Academy of Sciences* 97.12 (2000), pp. 6515–6520.

[88] Tor Ivry, Shahar Michal, Assaf Avihoo, Guillermo Sapiro, and Danny Barash. "An image processing approach to computing distances between RNA secondary structures dot plots." In: *Algorithms for Molecular Biology* 4.1 (2009), pp. 1–19.

[89] Luc Jaeger, Eric Westhof, and Neocles B Leontis. "TectoRNA: modular assembly units for the construction of RNA nano-objects." In: *Nucleic acids research* 29.2 (2001), pp. 455–463.

[90] Stefan Janssen and Robert Giegerich. "The RNA shapes studio." In: *Bioinformatics* (2015). DOI: 10.1093/bioinformatics/btu649. URL: http://bioinformatics.oxfordjournals.org/content/31/3/423.abstract.

[91] Martin Jinek, Alexandra East, Aaron Cheng, Steven Lin, Enbo Ma, and Jennifer Doudna. "RNA-programmed genome editing in human cells." In: *elife* 2 (2013), e00471.

[92] Steven G Johnson and Matteo Frigo. "A modified split-radix FFT with fewer arithmetic operations." In: *IEEE Transactions on Signal Processing* 55.1 (2006), pp. 111–119.

[93] Anis Farhan Kamaruzaman, Azlan Mohd Zain, Suhaila Mohamed Yusuf, and Amirmudin Udin. "Lévy flight algorithm for optimization problems—a literature review." In: *Applied Mechanics and Materials*. Vol. 421. Trans Tech Publ. 2013, pp. 496–501.

[94] Kazutaka Katoh, Kazuharu Misawa, Kei-ichi Kuma, and Takashi Miyata. "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform." In: *Nucleic Acids Research* 30.14 (2002), pp. 3059–3066.

[95] Michael Kertesz, Yue Wan, Elad Mazor, John L Rinn, Robert C Nutter, Howard Y Chang, and Eran Segal. "Genome-wide measurement of RNA secondary structure in yeast." In: *Nature* 467.7311 (2010), pp. 103–107.

[96] Yoon Ki Kim, Luc Furic, Marc Parisien, François Major, Luc DesGroseillers, and Lynne E Maquat. "Staufen1 regulates diverse classes of mammalian transcripts." In: *The EMBO journal* 26.11 (2007), pp. 2670–2681.

[97] Motoo Kimura. *The neutral theory of molecular evolution*. Cambridge University Press, 1983.

[98] Daniel J Klein, Thomas E Edwards, and Adrian R Ferré-D'Amaré. "Cocrystal structure of a class I preQ1 riboswitch reveals a pseudoknot recognizing an essential hypermodified nucleobase." In: *Nature structural & molecular biology* 16.3 (2009), pp. 343–344.

[99] Robert Kleinkauf, Torsten Houwaart, Rolf Backofen, and Martin Mann. "antaRNA–Multi-objective inverse folding of pseudoknot RNA using ant-colony optimization." In: *BMC bioinformatics* 16.1 (2015), pp. 1–7.

[100] Robert Kleinkauf, Martin Mann, and Rolf Backofen. "antaRNA: ant colony-based RNA sequence design." In: *Bioinformatics* 31.19 (2015), pp. 3114–3121.

[101] Konstantin Klemm, Christoph Flamm, and Peter F Stadler. "Funnels in energy landscapes." In: *The European Physical Journal B* 63.3 (2008), pp. 387–391.

[102] Bjarne Knudsen and Jotun Hein. "RNA secondary structure prediction using stochastic context-free grammars and evolutionary history." In: *Bioinformatics (Oxford, England)* 15.6 (1999), pp. 446–454.

[103] Bjarne Knudsen and Jotun Hein. "Pfold: RNA secondary structure prediction using stochastic context-free grammars." In: *Nucleic acids research* 31.13 (2003), pp. 3423–3428.

[104] Donald E. Knuth. "Computer Programming as an Art." In: *Communications of the ACM* 17.12 (1974), pp. 667–673.

[105] Rohan V Koodli, Benjamin Keep, Katherine R Coppess, Fernando Portela, Eterna participants, and Rhiju Das. "EternaBrain: Automated RNA design through move sets and strategies from an Internet-scale RNA videogame." In: *PLoS computational biology* 15.6 (2019), e1007059.

[106] Rohan V. Koodli, Boris Rudolfs, Hannah K. Wayment-Steele, Eterna Structure Designers, and Rhiju Das. "Redesigning the EteRNA100 for the Vienna 2 folding engine." In: *bioRxiv* (2021). DOI: 10.1101/2021.08.26.457839. eprint: https://www.biorxiv.org/content/early/2021/08/28/2021.08.26.457839.full.pdf. URL: https://www.biorxiv.org/content/early/2021/08/28/2021.08.26.457839.

[107] Marcel Kucharík, Ivo L. Hofacker, Peter F. Stadler, and Jing Qin. "Pseudoknots in RNA folding landscapes." In: *Bioinformatics* 32.2 (Jan. 15, 2016). Publisher: Oxford Academic, pp. 187–194. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btv572. URL: https://academic.oup.com/bioinformatics/article/32/2/187/1744549 (visited on 02/17/2021).

[108] Yingjun Li, Saifu Pan, Yan Zhang, Min Ren, Mingxia Feng, Nan Peng, Lanming Chen, Yun Xiang Liang, and Qunxin She. "Harnessing Type I and Type III CRISPR-Cas systems for genome editing." In: *Nucleic acids research* 44.4 (2016), e34–e34.

[109] Zhongsen Li, Zhan-Bin Liu, Aiqiu Xing, Bryan P Moon, Jessica P Koellhoffer, Lingxia Huang, R Timothy Ward, Elizabeth Clifton, S Carl Falco, and A Mark Cigan. "Cas9-guide RNA directed genome editing in soybean." In: *Plant physiology* 169.2 (2015), pp. 960–970.

[110] Adam Lipowski and Dorota Lipowska. "Roulette-wheel selection via stochastic acceptance." In: *Physica A: Statistical Mechanics and its Applications* 391.6 (2012), pp. 2193–2196.

[111] Qi Liu, Xiuzi Ye, and Yin Zhang. "A Hopfield neural network based algorithm for RNA secondary structure prediction." In: *First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*. Vol. 1. IEEE. 2006, pp. 10–16.

[112] Ronny Lorenz, Stephan H Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. "Viennarna Package 2.0." In: *Algorithms for Molecular Biology* 6.1 (2011), p. 26. DOI: 10.1186/1748-7188-6-26. URL: https://doi.org/10.1186/1748-7188-6-26.

[113] Ronny Lorenz, Stephan H Bernhart, Christian Höner Zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. "ViennaRNA Package 2.0." In: *Algorithms for molecular biology* 6.1 (2011), p. 26.

[114] Ronny Lorenz, Christoph Flamm, Ivo Hofacker, and Peter Stadler. "Efficient computation of base-pairing probabilities in multi-strand RNA folding." In: *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies*. 2020, pp. 23–31. DOI: 10.5220/0008916600230031. URL: https://doi.org/10.5220/0008916600230031.

[115] Ronny Lorenz, Michael T Wolfinger, Andrea Tanzer, and Ivo L Hofacker. "Predicting RNA secondary structures from sequence and probing data." In: *Methods* 103 (2016), pp. 86–98.

[116] Zhi John Lu, Jason W Gloor, and David H Mathews. "Improved RNA secondary structure prediction by maximizing expected pair accuracy." In: *Rna* 15.10 (2009), pp. 1805–1813.

[117] Rune B Lyngsø, James WJ Anderson, Elena Sizikova, Amarendra Badugu, Tomas Hyland, and Jotun Hein. "Frnakenstein: multiple target inverse RNA folding." In: *BMC bioinformatics* 13.1 (2012), pp. 1–12.

[118] Rune B. Lyngsø and Christian N. S. Pedersen. "RNA Pseudoknot Prediction in Energy-Based Models." In: *Journal of Computational Biology* 7.3 (Aug. 1, 2000). Publisher: Mary Ann Liebert, Inc., publishers, pp. 409–427. DOI: 10.1089/106652700750050862. URL: https://www.liebertpub.com/doi/10.1089/106652700750050862 (visited on 04/16/2021).

[119] JT Madison, GA Everett, and H Kung. "Nucleotide sequence of a yeast tyrosine transfer RNA." In: *Science* 153.3735 (1966), pp. 531–534.

[120] B Mandelbrot. "Certain speculative prices (1963)." In: *The Journal of Business* 45.4 (1972), pp. 542–543.

[121] Hugo M. Martinez. "An RNA folding rule." In: *Nucleic Acids Research* 12.1 (1984), pp. 323–334. DOI: 10.1093/nar/12.1part1.323. URL: https://doi.org/10.1093/nar/12.1part1.323.

[122] David H. Mathews. "How to benchmark RNA secondary structure prediction accuracy." In: *Methods* 162-163.162 (2019), pp. 60–67. DOI: 10.1016/j.ymeth.2019.04.003. URL: https://doi.org/10.1016/j.ymeth.2019.04.003.

[123] David H Mathews, Matthew D Disney, Jessica L Childs, Susan J Schroeder, Michael Zuker, and Douglas H Turner. "Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure." In: *Proceedings of the National Academy of Sciences* 101.19 (2004), pp. 7287–7292.

[124] David H Mathews, Jeffrey Sabina, Michael Zuker, and Douglas H Turner. "Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure." In: *Journal of Molecular Biology* 288.5 (1999), pp. 911–940.

[125] DH Matthews, TC Andre, J Kim, DH Turner, and M Zuker. "An updated recursive algorithm for RNA secondary structure prediction with improved thermodynamic parameters." In: (1998).

[126] Marco C Matthies, Stefan Bienert, and Andrew E Torda. "Dynamics in sequence space for RNA secondary structure design." In: *Journal of chemical theory and computation* 8.10 (2012), pp. 3663–3670.

[127] John S McCaskill. "The equilibrium partition function and base pair binding probabilities for RNA secondary structure." In: *Biopolymers: Original Research on Biomolecules* 29.6-7 (1990), pp. 1105–1119.

[128] Nono SC Merleau and Matteo Smerlak. "A simple evolutionary algorithm guided by local mutations for an efficient RNA design." In: *Proceedings of the Genetic and Evolutionary Computation Conference*. 2021, pp. 1027–1034.

[129] Nono SC Merleau and Matteo Smerlak. "An evolutionary algorithm for inverse RNA folding inspired by Lévy flights." In: *bioRxiv* (2022).

[130] Gerard Minuesa, Cristina Alsina, Juan Antonio Garcia-Martin, Juan Carlos Oliveros, and Ivan Dotu. "MoiRNAiFold: a novel tool for complex in silico RNA design." In: *Nucleic acids research* 49.9 (2021), pp. 4934–4943.

[131] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[132] Soheila Montaseri, Mohammad Ganjtabesh, and Fatemeh Zare-Mirakabad. "Evolutionary algorithm for RNA secondary structure prediction based on simulated SHAPE data." In: *PloS one* 11.11 (2016), e0166965.

[133] Peter B Moore and Thomas A Steitz. "The roles of RNA in the synthesis of protein." In: *Cold Spring Harbor perspectives in biology* 3.11 (2011), a003780.

[134] Steffen Mueller, J Robert Coleman, Dimitris Papamichail, Charles B Ward, Anjaruwee Nimnual, Bruce Futcher, Steven Skiena, and Eckard Wimmer. "Live attenuated influenza virus vaccines by computer-aided rational design." In: *Nature biotechnology* 28.7 (2010), pp. 723–726.

[135] Mark EJ Newman. "Power laws, Pareto distributions and Zipf's law." In: *Contemporary physics* 46.5 (2005), pp. 323–351.

[136] Ruth Nussinov and Ann B Jacobson. "Fast algorithm for predicting the secondary structure of single-stranded RNA." In: *Proceedings of the National Academy of Sciences* 77.11 (1980), pp. 6309–6313.

[137] Bibiana Onoa and Ignacio Tinoco Jr. "RNA folding and unfolding." In: *Current opinion in structural biology* 14.3 (2004), pp. 374–379.

[138] Vaitea Opuu, Nono SC Merleau, Messow Vincent, and Matteo Smerlak. "RAFFT: Efficient prediction of RNA folding pathways using the fast Fourier transform." In: *bioRxiv* (2021).

[139] Jie Pan, D. Thirumalai, and Sarah A. Woodson. "Folding of RNA involves parallel pathways." In: *Journal of Molecular Biology* 273.1 (1997), pp. 7–13. DOI: 10.1006/jmbi.1997.1311. URL: https://doi.org/10.1006/jmbi.1997.1311.

[140] Marc Parisien and Francois Major. "The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data." In: *Nature* 452.7183 (2008), pp. 51–55.

[141] Fernando Portela. "An unexpectedly effective Monte Carlo technique for the RNA inverse folding problem." 2018.

[142] "RNAcentral: a comprehensive database of non-coding RNA sequences." In: *Nucleic acids research* 45.D1 (2017), pp. D128–D134.

[143] Effirul I Ramlan and Klaus-Peter Zauner. "Design of interacting multi-stable nucleic acids for molecular information processing." In: *Biosystems* 105.1 (2011), pp. 14–24.

[144] Jens Reeder, Peter Steffen, and Robert Giegerich. "pknotsRG: RNA pseudoknot folding including near-optimal structures and sliding windows." In: *Nucleic acids research* 35.suppl_2 (2007), W320–W324.

[145] Christian Reidys, Peter F Stadler, and Peter Schuster. "Generic properties of combinatory maps: neutral networks of RNA secondary structures." In: *Bulletin of mathematical biology* 59.2 (1997), pp. 339–397.

[146] Vladimir Reinharz, Yann Ponty, and Jérôme Waldispühl. "A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution." In: *Bioinformatics* 29.13 (2013), pp. i308–i315. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt217. eprint: https://academic.oup.com/bioinformatics/article-pdf/29/13/i308/18534655/btt217.pdf. URL: https://doi.org/10.1093/bioinformatics/btt217.

[147] Jihong Ren, Baharak Rastegari, Anne Condon, and Holger H Hoos. "HotKnots: heuristic prediction of RNA secondary structures including pseudoknots." In: *RNA* 11.10 (2005), pp. 1494–1504.

[148] Jessica S Reuter and David H Mathews. "RNAstructure: software for RNA secondary structure prediction and analysis." In: *BMC bioinformatics* 11.1 (2010), pp. 1–9.

[149] Andy M Reynolds. "Current status and future directions of Lévy walk research." In: *Biology open* 7.1 (2018), bio030106.

[150] Elena Rivas and Sean R. Eddy. "A dynamic programming algorithm for RNA structure prediction including pseudoknots." In: *Journal of Molecular Biology* 285.5 (1999), pp. 2053–2068. ISSN: 0022-2836. DOI: https://doi.org/10.1006/jmbi.1998.2436. URL: https://www.sciencedirect.com/science/article/pii/S0022283698924366.

[151] Elena Rivas, Raymond Lang, and Sean R Eddy. "A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more." In: *RNA* 18.2 (2012), pp. 193–212.

[152] Elena Rivas, Raymond Lang, and Sean R Eddy. "A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more." In: *RNA* 18.2 (2012), pp. 193–212.

[153] Debra L Robertson and Gerald F Joyce. "Selection in vitro of an RNA enzyme that specifically cleaves single-stranded DNA." In: *Nature* 344.6265 (1990), pp. 467–468.

[154] John M Rosenberg, Nadrian C Seeman, Roberta O Day, and Alexander Rich. "RNA double-helical fragments at atomic resolution: II. The crystal structure of sodium guanylyl-3′, 5′-cytidine nonahydrate." In: *Journal of molecular biology* 104.1 (1976), pp. 145–167.

[155] Frederic Runge, Danny Stoll, Stefan Falkner, and Frank Hutter. "Learning to design RNA." In: *arXiv preprint arXiv:1812.11951* (2018).

[156] Rick Russell, Xiaowei Zhuang, Hazen P Babcock, Ian S Millett, Sebastian Doniach, Steven Chu, and Daniel Herschlag. "Exploring the folding landscape of a structured RNA." In: *Proceedings of the National Academy of Sciences* 99.1 (2002), pp. 155–160.

[157] Yasubumi Sakakibara, Michael Brown, Richard Hughey, I Saira Mian, Kimmen Sjölander, Rebecca C Underwood, and David Haussler. "Stochastic context-free grammers for tRNA modeling." In: *Nucleic acids research* 22.23 (1994), pp. 5112–5120.

[158] Tore Samuelsson and Christian Zwieb. "The signal recognition particle database (SRPDB)." In: *Nucleic Acids Research* 27.1 (1999), pp. 169–170.

[159] Baby Santosh, Akhil Varshney, and Pramod Kumar Yadava. "Non-coding RNAs: biological functions and applications." In: *Cell biochemistry and function* 33.1 (2015), pp. 14–22.

[160] Kengo Sato, Manato Akiyama, and Yasubumi Sakakibara. "RNA secondary structure prediction using deep learning with thermodynamic integration." In: *Nature communications* 12.1 (2021), pp. 1–9.

[161] Kengo Sato, Manato Akiyama, and Yasubumi Sakakibara. *RNA secondary structure prediction using deep learning with thermodynamic integration*. 2021.

[162] Kengo Sato, Yuki Kato, Michiaki Hamada, Tatsuya Akutsu, and Kiyoshi Asai. "IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming." In: *Bioinformatics* 27.13 (2011), pp. i85–i93.

[163] Martin Sauvageau, Loyal A Goff, Simona Lodato, Boyan Bonev, Abigail F Groff, Chiara Gerhardinger, Diana B Sanchez-Gomez, Ezgi Hacisuleyman, Eric Li, Matthew Spence, et al. "Multiple knockout mouse models reveal lincRNAs are required for life and brain development." In: *elife* 2 (2013), e01749.

[164] Murray N Schnare, Simon H Damberger, Michael W Gray, and Robin R Gutell. "Comprehensive comparison of structural characteristics in eukaryotic cytoplasmic large subunit (23 S-like) ribosomal RNA." In: *Journal of Molecular Biology* 256.4 (1996), pp. 701–719.

[165] Peter Schuster, Walter Fontana, Peter F Stadler, and Ivo L Hofacker. "From sequences to shapes and back: a case study in RNA secondary structures." In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 255.1344 (1994), pp. 279–284.

[166] Nadrian C Seeman, John M Rosenberg, FL Suddath, Jung Ja Park Kim, and Alexander Rich. "RNA double-helical fragments at atomic resolution: I. The crystal and molecular structure of sodium adenylyl-3′, 5′-uridine hexahydrate." In: *Journal of molecular biology* 104.1 (1976), pp. 109–144.

[167] Matthew G Seetin and David H Mathews. "RNA structure prediction: an overview of methods." In: *Bacterial Regulatory RNA* (2012), pp. 99–122.

[168] Martin J Serra and Douglas H Turner. "Predicting thermodynamic properties of RNA." In: *Methods in enzymology*. Vol. 259. Elsevier, 1995, pp. 242–261.

[169] Bruce A Shapiro and Kaizhong Zhang. "Comparing multiple RNA secondary structures using tree comparisons." In: *Bioinformatics* 6.4 (1990), pp. 309–318.

[170] Vishnu Prakash Sharma, Harji Ram Choudhary, Sandeep Kumar, and Vikas Choudhary. "A modified DE: Population or generation based levy flight differential evolution (PGLFDE)." In: *2015 International Conference on Futuristic*

Trends on Computational Analysis and Knowledge Management (ABLAZE). IEEE. 2015, pp. 704–710.

[171] Jade Shi, Rhiju Das, and Vijay S Pande. "SentRNA: Improving computational RNA design by incorporating a prior of human design strategies." 2018.

[172] Micheal F Shlesinger, George M Zaslavsky, and Uriel Frisch. *Lévy flights and related topics in physics*. Vol. 450. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995.

[173] Wenjie Shu, Ming Liu, Hebing Chen, Xiaochen Bo, and Shengqi Wang. "ARDesigner: a web-based system for allosteric RNA design." In: *Journal of biotechnology* 150.4 (2010), pp. 466–473.

[174] Christian Höner zu Siederdissen, Stephan H Bernhart, Peter F Stadler, and Ivo L Hofacker. "A folding algorithm for extended RNA secondary structures." In: *Bioinformatics* 27.13 (2011), pp. i129–i136.

[175] Jaswinder Singh, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. "RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning." In: *Nature Communications* 10.1 (2019), pp. 1–13.

[176] Michael F Sloma and David H Mathews. "Exact calculation of loop formation probability identifies folding motifs in RNA secondary structures." In: *RNA* 22.12 (2016), pp. 1808–1818.

[177] Michael F Sloma and David H Mathews. "Base pair probability estimates improve the prediction accuracy of RNA non-canonical base pairs." In: *PLoS computational biology* 13.11 (2017), e1005827.

[178] Matteo Smerlak. "Effective potential reveals evolutionary trajectories in complex fitness landscapes." In: *arXiv preprint arXiv:1912.05890* (2019).

[179] Matteo Smerlak. "Neutral quasispecies evolution and the maximal entropy random walk." In: *Science advances* 7.16 (2021), eabb2376.

[180] Sergey V. Solomatin, Max Greenfeld, Steven Chu, and Daniel Herschlag. "Multiple native states reveal persistent ruggedness of an RNA folding landscape." In: *Nature* 463.7281 (2010), pp. 681–684. DOI: 10.1038/nature08717. URL: https://doi.org/10.1038/nature08717.

[181] T. Specht, M. Szymanski, M. Z. Barciszewska, J. Barciszewski, and V. A. Erdmann. "Compilation of 5s rRNA and 5s rRNA gene sequences." In: *Nucleic Acids Research* 25.1 (1997), pp. 96–97. DOI: 10.1093/nar/25.1.96. URL: https://doi.org/10.1093/nar/25.1.96.

[182] Robert C Spitale, Andrew T Torelli, Jolanta Krucinska, Vahe Bandarian, and Joseph E Wedekind. "The Structural Basis for Recognition of the PreQo Metabolite by an Unusually Small Riboswitch Aptamer Domain." In: *Journal of Biological Chemistry* 284.17 (2009), pp. 11012–11016.

[183] Mathias Sprinzl, Carsten Horn, Melissa Brown, Anatoli Ioudovitch, and Sergey Steinberg. "Compilation of tRNA sequences and sequences of tRNA genes." In: *Nucleic Acids Research* 26.1 (1998), pp. 148–153.

[184] Evan W Steeg. "Neural networks, adaptive optimization, and RNA secondary structure prediction." In: *Artificial intelligence and molecular biology* (1993), pp. 121–60.

[185] Paul R Stein and Michael S Waterman. "On some new sequences generalizing the Catalan and Motzkin numbers." In: *Discrete Mathematics* 26.3 (1979), pp. 261–272.

[186] Sergei Svitashev, Joshua K Young, Christine Schwartz, Huirong Gao, S Carl Falco, and A Mark Cigan. "Targeted mutagenesis, precise gene editing, and site-specific gene insertion in maize using Cas9 and guide RNA." In: *Plant physiology* 169.2 (2015), pp. 931–945.

[187] Yoshiyasu Takefuji and L Chen. "Parallel algorithms for finding a near-maximum independent set of." In: *IEEE Trans. Neural Networks* 1.3 (1990), p. 263.

[188] Akito Taneda. "MODENA: a multi-objective RNA inverse folding." In: *Advances and applications in bioinformatics and chemistry: AABC* 4 (2011), p. 1.

[189] Akito Taneda. "Multi-Objective Genetic Genetic for Pseudoknotted RNA Sequence Design." In: *Frontiers in Genetics* 3 (2012), p. 36. ISSN: 1664-8021. DOI: 10.3389/fgene.2012.00036. URL: https://www.frontiersin.org/article/10.3389/fgene.2012.000INFO-RNA36.

[190] Akito Taneda. "Multi-objective optimization for RNA design with multiple target secondary structures." In: *BMC bioinformatics* 16.1 (2015), pp. 1–20.

[191] Michela Taufer, Abel Licon, Roberto Araiza, David Mireles, FHD Van Batenburg, Alexander P Gultyaev, and Ming-Ying Leung. "PseudoBase++: an extension of PseudoBase for easy searching, formatting and visualization of pseudoknots." In: *Nucleic acids research* 37.suppl_1 (2009), pp. D127–D135.

[192] Siqi Tian and Rhiju Das. "RNA structure through multidimensional chemical mapping." In: *Quarterly reviews of biophysics* 49 (2016).

[193] Pilar Tijerina, Sabine Mohr, and Rick Russell. "DMS footprinting of structured RNAs and RNA–protein complexes." In: *Nature protocols* 2.10 (2007), pp. 2608–2623.

[194] Ignacio Tinoco Jr and Carlos Bustamante. "How RNA folds." In: *Journal of molecular biology* 293.2 (1999), pp. 271–281.

[195] Ignacio Tinoco, Olke C. Uhlenbeck, and Mark D. Levine. "Estimation of Secondary Structure in Ribonucleic Acids." In: *Nature* 230.5293 (Apr. 1971). Number: 5293 Publisher: Nature Publishing Group, pp. 362–367. ISSN: 1476-4687. DOI: 10.1038/230362a0. URL: https://www.nature.com/articles/230362a0 (visited on 04/14/2021).

[196] Craig Tuerk and Larry Gold. "Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase." In: *science* 249.4968 (1990), pp. 505–510.

[197] Douglas H. Turner and David H. Mathews. "NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure." In: *Nucleic Acids Research* 38.suppl1 (2009), pp. D280–D282.

[198] Douglas H Turner and David H Mathews. "NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure." In: *Nucleic Acids Research* 38.suppl_1 (2010), pp. D280–D282.

[199] Sinan Uğur Umu and Paul P Gardner. "A comprehensive benchmark of RNA–RNA interaction prediction tools for all domains of life." In: *Bioinformatics* 33.7 (2017), pp. 988–996.

[200] Jason G Underwood, Andrew V Uzilov, Sol Katzman, Courtney S Onodera, Jacob E Mainzer, David H Mathews, Todd M Lowe, Sofie R Salama, and David Haussler. "FragSeq: transcriptome-wide RNA structure probing using high-throughput sequencing." In: *Nature methods* 7.12 (2010), pp. 995–1001.

[201] Gandhimohan M Viswanathan, EP Raposo, and MGE Da Luz. "Lévy flights and superdiffusion in the context of biological encounters and random searches." In: *Physics of Life Reviews* 5.3 (2008), pp. 133–150.

[202] Alexey G Vitreschak, Dimitry A Rodionov, Andrey A Mironov, and Mikhail S Gelfand. "Riboswitches: the oldest mechanism for the regulation of gene expression?" In: *Trends in Genetics* 20.1 (2004), pp. 44–50.

[203] Manja Wachsmuth, Gesine Domin, Ronny Lorenz, Robert Serfling, Sven Findeiß, Peter F Stadler, and Mario Mörl. "Design criteria for synthetic riboswitches acting on transcription." In: *RNA biology* 12.2 (2015), pp. 221–231.

[204] Haoyi Wang, Hui Yang, Chikdu S Shivalila, Meelad M Dawlaty, Albert W Cheng, Feng Zhang, and Rudolf Jaenisch. "One-step generation of mice carrying mutations in multiple genes by CRISPR/Cas-mediated genome engineering." In: *cell* 153.4 (2013), pp. 910–918.

[205] Shouhua Wang, Ting ting Su, Huanjun Tong, Weibin Shi, Fei Ma, and Zhiwei Quan. "CircPVT1 promotes gallbladder cancer growth by sponging miR-339-3p and regulates MCL-1 expression." In: *Cell Death Discovery* 7.1 (2021), pp. 1–10.

[206] Zhong Wang, Mark Gerstein, and Michael Snyder. "RNA-Seq: a revolutionary tool for transcriptomics." In: *Nature reviews genetics* 10.1 (2009), pp. 57–63.

[207] Richard B Waring and R Wayne Davies. "Assessment of a model for intron RNA secondary structure relevant to RNA self-splicing—a review." In: *Gene* 28.3 (1984), pp. 277–291.

[208] James D Watson and Francis HC Crick. "Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid." In: *Nature* 171.4356 (1953), pp. 737–738.

[209] Lina Weinbrand, Assaf Avihoo, and Danny Barash. "RNAfbinv: an interactive Java application for fragment-based design of RNA sequences." In: *Bioinformatics* 29.22 (2013), pp. 2938–2940.

[210] Eric Westhof and Valérie Fritsch. "RNA folding: beyond Watson–Crick pairs." In: *Structure* 8.3 (2000), R55–R65. ISSN: 0969-2126. DOI: https://doi.org/10.1016/S0969-2126(00)00112-X. URL: https://www.sciencedirect.com/science/article/pii/S096921260000112X.

[211] Kay C Wiese, Andrew Hendriks, and Jagdeep Poonian. "Algorithms for RNA folding: a comparison of dynamic programming and parallel evolutionary algorithms." In: *2005 IEEE Congress on Evolutionary Computation*. Vol. 1. IEEE. 2005, pp. 475–483.

[212] Kevin A Wilkinson, Edward J Merino, and Kevin M Weeks. "Selective 2′-hydroxyl acylation analyzed by primer extension (SHAPE): quantitative RNA structure analysis at single nucleotide resolution." In: *Nature protocols* 1.3 (2006), pp. 1610–1616.

[213] Wade C Winkler and Ronald R Breaker. "Genetic control by metabolite-binding riboswitches." In: *Chembiochem* 4.10 (2003), pp. 1024–1032.

[214] SA Woodson. "Recent insights on RNA folding mechanisms from catalytic RNA." In: *Cellular and Molecular Life Sciences CMLS* 57.5 (2000), pp. 796–808.

[215] Xiufeng Yang, Kazuki Yoshizoe, Akito Taneda, and Koji Tsuda. "RNA inverse folding using Monte Carlo tree search." In: *BMC bioinformatics* 18.1 (2017), p. 468.

[216] Hua-Ting Yao, Cedric Chauve, Mireille Regnier, and Yann Ponty. "Exponentially few RNA structures are designable." In: *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 2019, pp. 289–298.

[217] Hua-Ting Yao, Jérôme Waldispühl, Yann Ponty, and Sebastian Will. "Taming Disruptive Base Pairs to Reconcile Positive and Negative Structural Design of RNA." In: *RECOMB 2021-25th international conference on research in computational molecular biology*. 2021.

[218] Joseph N Zadeh, Conrad D Steenberg, Justin S Bois, Brian R Wolfe, Marshall B Pierce, Asif R Khan, Robert M Dirks, and Niles A Pierce. "NUPACK: Analysis and design of nucleic acid systems." In: *Journal of computational chemistry* 32.1 (2011), pp. 170–173.

[219] Joseph N Zadeh, Brian R Wolfe, and Niles A Pierce. "Nucleic acid sequence design via efficient ensemble defect optimization." In: *Journal of computational chemistry* 32.3 (2011), pp. 439–452.

[220] Shay Zakov, Yoav Goldberg, Michael Elhadad, and Michal Ziv-Ukelson. "Rich parameterization improves RNA structure prediction." In: *Journal of Computational Biology* 18.11 (2011), pp. 1525–1542.

[221] Wenbing Zhang and Shi-Jie Chen. "RNA hairpin-folding kinetics." In: *Proceedings of the National Academy of Sciences* 99.4 (2002), pp. 1931–1936.

[222] Wenbing Zhang and Shi-Jie Chen. "Analyzing the biopolymer folding rates and pathways using kinetic cluster method." In: *The Journal of chemical physics* 119.16 (2003), pp. 8716–8729.

[223] Wenbing Zhang and Shi-Jie Chen. "Exploring the complex folding kinetics of RNA hairpins: I. General folding kinetics analysis." In: *Biophysical journal* 90.3 (2006), pp. 765–777.

[224] Wenbing Zhang and Shi-Jie Chen. "Exploring the complex folding kinetics of RNA hairpins: I. General folding kinetics analysis." In: *Biophysical Journal* 90.3 (2006), pp. 765–777.

[225] Qi Zhao, Zheng Zhao, Xiaoya Fan, Zhengwei Yuan, Qian Mao, and Yudong Yao. "Review of machine learning methods for RNA secondary structure prediction." In: *PLoS computational biology* 17.8 (2021), e1009291.

[226] Yu Zhu, ZhaoYang Xie, YiZhou Li, Min Zhu, and Yi-Ping Phoebe Chen. "Research on folding diversity in statistical learning methods for RNA secondary structure prediction." In: *International Journal of Biological Sciences* 14.8 (2018), p. 872.

[227] Michael Zuker and David Sankoff. "RNA secondary structures and their prediction." In: *Bulletin of mathematical biology* 46.4 (1984), pp. 591–621.

[228] Michael Zuker and Patrick Stiegler. "Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information." In: *Nucleic acids research* 9.1 (1981), pp. 133–148.

[229] C. Zwieb. "Tmrdb (tmRNA database)." In: *Nucleic Acids Research* 28.1 (2000), pp. 169–170. DOI: 10.1093/nar/28.1.169. URL: https://doi.org/10.1093/nar/28.1.169.

[230] C. Zwieb. "Tmrdb (tmRNA database)." In: *Nucleic Acids Research* 31.1 (2003), pp. 446–447. DOI: 10.1093/nar/gkg019. URL: https://doi.org/10.1093/nar/gkg019.

DECLARATION

Put your declaration here.

Leipzig, June 2022

Nono Saha Cyrille Merleau

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/

Thank you very much for your feedback and contribution.