From RNA folding to inverse folding:
a heuristic exploration

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 Diplommathematiker **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 Micheal Saha.

ABSTRACT

TODEO: Write the abstract here.

`https://plg.uwaterloo.ca/~migod/research/beckOOPSLA.html`

ZUSAMMENFASSUNG

Kurze Zusammenfassung des Inhaltes in deutscher Sprache...

PUBLICATIONS

This is just an early – and currently ugly – test!

This might come in handy for PhD theses: some ideas and figures have appeared previously in the following publications:

Attention: This requires a separate run of `bibtex` for your `refsection`, e. g., `ClassicThesis1-blx` for this file. You might also use `biber` as the backend for `biblatex`. See also `http://tex.stackexchange.com/questions/128196/problem-with-refsection`.

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 [13]

ACKNOWLEDGMENTS

Put your acknowledgments here.

Many thanks to everybody who already sent me a postcard!

Regarding the typography and other help, many thanks go to Marco Kuhlmann, Philipp Lehman, Lothar Schlesier, Jim Young, Lorenzo Pantieri and Enrico Gregorio[1], Jörg Sommer, Joachim Köstler, Daniel Gottschlag, Denis Aydin, Paride Legovini, Steffen Prochnow, Nicolas Repp, Hinrich Harms, Roland Winkler, Jörg Weber, Henri Menke, Claus Lahiri, Clemens Niederberger, Stefano Bragaglia, Jörn Hees, Scott Lowe, Dave Howcroft, José M. Alcaide, David Carlisle, Ulrike Fischer, Hugues de Lassus, Csaba Hajdu, Dave Howcroft, and the whole LaTeX-community for support, ideas and some great software.

Regarding LyX: The LyX port was intially done by *Nicholas Mariette* in March 2009 and continued by *Ivo Pletikosić* in 2011. Thank you very much for your work and for the contributions to the original style.

1 Members of GuIT (Gruppo Italiano Utilizzatori di TeX e LaTeX)

CONTENTS

0 PREFACE 1

1 INTRODUCTION TO NUCLEIC ACID STRUCTURE 3
- 1.1 Survey 3
- 1.2 Deoxy-nucleic Acids (DNAs) 3
- 1.3 non-coding RNAs and their biological implications 4
  - 1.3.1 Biochemistry of RNA molecules 5
  - 1.3.2 Biological function of non-coding RNAs 7
- 1.4 Bioinformatic definitions and RNA concepts. 9

I RNA FOLDING

2 RNA FOLDING 17
- 2.1 From RNA sequences to RNA structures 17
- 2.2 Stability and prediction of RNA secondary structures 17
- 2.3 Energy landscape and RNA kinetics 18
- 2.4 A literature review of RNA folding tools. 18
  - 2.4.1 Exact MFE prediction methods 18
  - 2.4.2 Statical methods 18
  - 2.4.3 Heuristic methods 18
- 2.5 Conclusion 19

3 RAFFT: EFFICIENT PREDICTION OF FAST-FOLDING PATHWAYS OF RNAS 21
- 3.1 Material and Methods 21
  - 3.1.1 Folding algorithm 21
  - 3.1.2 Kinetic ansatz 25
  - 3.1.3 Benchmark dataset used and structure prediction protocol. 26
  - 3.1.4 Structure space visualization 27
- 3.2 Experimental results 28
  - 3.2.1 Application to the folding task 28
  - 3.2.2 Selected applications of the kinetic ansatz 30
- 3.3 Conclusion 33

II RNA DESIGN

4 RNA DESIGN 39
- 4.1 Biological motivation and biotechnological implications 39
- 4.2 Positive and negative design. 39

4.3 Objective functions previously used. 39
4.4 A review on existing inverse RNA folding tools. 39
4.4.1 Pseudoknot-free RNA inverse folding tools 39
4.4.2 Pseudoknotted RNA inverse folding tools 41
4.5 Conclusion 41
5 arnaque: AN EVOLUTIONARY ALGORITHM FOR INVERSE FOLDING INSPIRED BY LÉVY FLIGHTS. 43
5.1 Material and methods 43
5.1.1 Inverse folding evolutionary algorithm (EA) 43
5.1.2 Parameter analysis and benchmark 45
5.1.3 Benchmark protocol 47
5.2 Experimental results 49
5.2.1 Performance on PseudoBase++: Levy mutation *vs.* Local mutation 50
5.2.2 Performance on PseudoBase++: aRNAque *vs.* antaRNA 51
5.2.3 Performance on Eterna100 dataset 52
5.3 Conclusion 53
III DISCUSSION AND PERSPECTIVE
6 RAFFT AND CONTINUOUS TRANSITION IN EVOLUTION 61
IV APPENDIX
BIBLIOGRAPHY 65

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

0 PREFACE

The preface will contain three paragraphs as follows:

A short story on the initial questions, and what this thesis is focused on.

Chronology on how we have arrived the main question of this thesis.

The research question question we address in this work

The outline of the thesis

1

INTRODUCTION TO NUCLEIC ACID STRUCTURE

1.1 SURVEY

In Biology, organisms are the living entities that consist of organs and the organs are made of tissues. The fundamental building blocks of the organisms that form tissues are cells. Cells consist of various molecules and molecular biology is the field of science that studies cells at the molecular level.

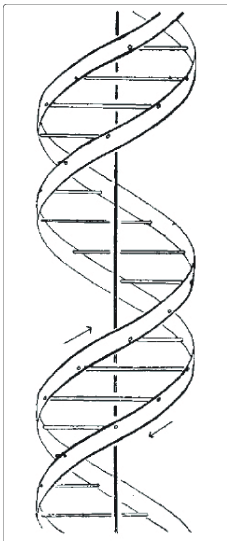
Molecules in the cell participate in various biochemical reactions to maintain the proper structure and function of the cell. There are micromolecules that are small molecules of low weights, they often refer to monomers. Many micromolecules can join together to form more complex molecules called macromolecules. There are four essential macromolecules in the cell—the nucleic acids that carry the genetics blueprint and the instructions for the functioning of the cell—the lipids that—the proteins that are one of the most abundant organic molecules in the living systems. They contribute in many functional activities: enzymatic catalyses, contractility, formation of selectively permeable membranes, reversible binding and transport, and immunological activities.—the carbohydrates that are essential part of our diet. They provide energy to the body; grains, fruits and vegetable are all considered to be natural source of carbohydrates.

Our work focuses on the nucleic acids. The nucleic acids are made up of small repetitive micromolecules called nucleotides. Genetic information necessary to specify the proteins needed by the organisms is contained in a nucleic acid call deoxyribonucleic acid (DNA) or, in some cases for some viruses in the ribonucleic acid (RNA).

1.2 DEOXY-NUCLEIC ACIDS (DNAS)

DNAs are macromolecules contained 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, each of them distinguished by the nucleobase they have: A for Ade-

nine , T for Thymine, G for Guanine, and C for Cytosine. Even though the basis blocks constituting the DNA was known for many years, it was only in 1953 that James Watson and Franklin Crick succeeded in putting them together and suggested a reasonable DNA structure. Their work relayed on DNA X-ray diffraction pattern produced by Rosalind Franklin and Maurice Wilking and the data from Erwin Charganp. 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 purimide and pyrimidine bases, they are joined together in pairs, a singlle base from the other chain bonded to a single base from the other chain. For the bonding to occur one of the pair must be Ademine and thymine or Guanine and Cytosine. The complementary pairing of the bases was then compatible with Chargaff's empirical rules—the amount of pyrimidine nucleotides (T+C) always equals the total number of purine nucleotides (A+G).



Helical representation of DNA structures.

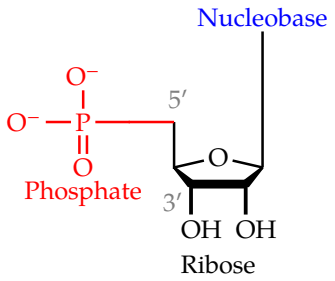
The elucidation of DNA structure by Watson and Crick has then motivated many other scientists for futher investigations and gave rise to a modern molecular biology. Later in the same year, Watson and Crick formulated the central dogma of molecular biology that describes the flow of information between DNA, RNA, and proteins. Fig 2 illustrates the dogma in two steps: from DNA to mRNA through transcription, from mRNAs to proteins through translation. Since this central dogma was proposed, more works have been done in investigating in details each steps of the Fig 2. DNAs are transcribed into RNA molecules (messenger RNAs) that contain the same information as the template DNAs, and subsequently these RNA messengers are translated into proteins according to the genetic code [Miller et al., 2009].

1.3 NON-CODING RNAS AND THEIR BIOLOGICAL IMPLICATIONS

So far in this work, RNAs have been considered as paying the same common role as DNAs which is the genetic information memory. But not all RNAs are translated into proteins, in other terms not all RNAs are mRNAs. There are mainly two RNA's groups: coding RNAs (cRNAs) that are translated into proteins and non-coding RNAs that are not translated into proteins. In the previous section, we introduced the central dogma of microbilogy which describe the flow of informations in the living systems. In other terms information flows from nucleic acids to proteins and not vice versa. It therefore, appears that DNAs and

proteins are vital components of living systems. But during the transcription and the translation steps in the information flow, there are some functions performed by non-coding RNAs such as ribosomal RNA (rRNAs) and transfer RNAs (tRNAs). The study of such RNAs revealed that rRNAs rather than ribosomal proteins catalyse 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. [Ref Moore PB, Steiz TA. The role of RNA in the synthesis of proteins. In: Gesteland RF, Cech TR, Atkins JF, editors. The RNA world. 3rd edition. Cold Spring Harbor (NY, USA): Cold Spring Harbor Laboratory Press; 2005. p. 257–85.]. Such studies suggest that RNAs are the actual catalysts of protein synthesis even though proteins remain the common catalysts of various chemical reactions occurring in the cell. This implies also that not only proteins, but also RNAs can function as efficient catalysts. In addition to protein synthesis, experiments in vitro evolution have shown that RNA molecules can catalyse a variety of chemical reactions relevant to biological processes such as RNA replication, nucleotide synthesis, thymidylate synthesis, lipid synthesis, and sugar metabolism (see Robertson DL, Joyce GF. Selection in vitro of an RNA enzyme that specifically cleaves single-stranded DNA. Nature , Ellington AD, Chen X, Robertson M, Syrett A. Evolutionary origins and directed evolution of RNA. The International Journal of Biochem- istry & Cell Biology). Therefore, RNA molecules can perform functions partially equivalent to those performed by proteins.

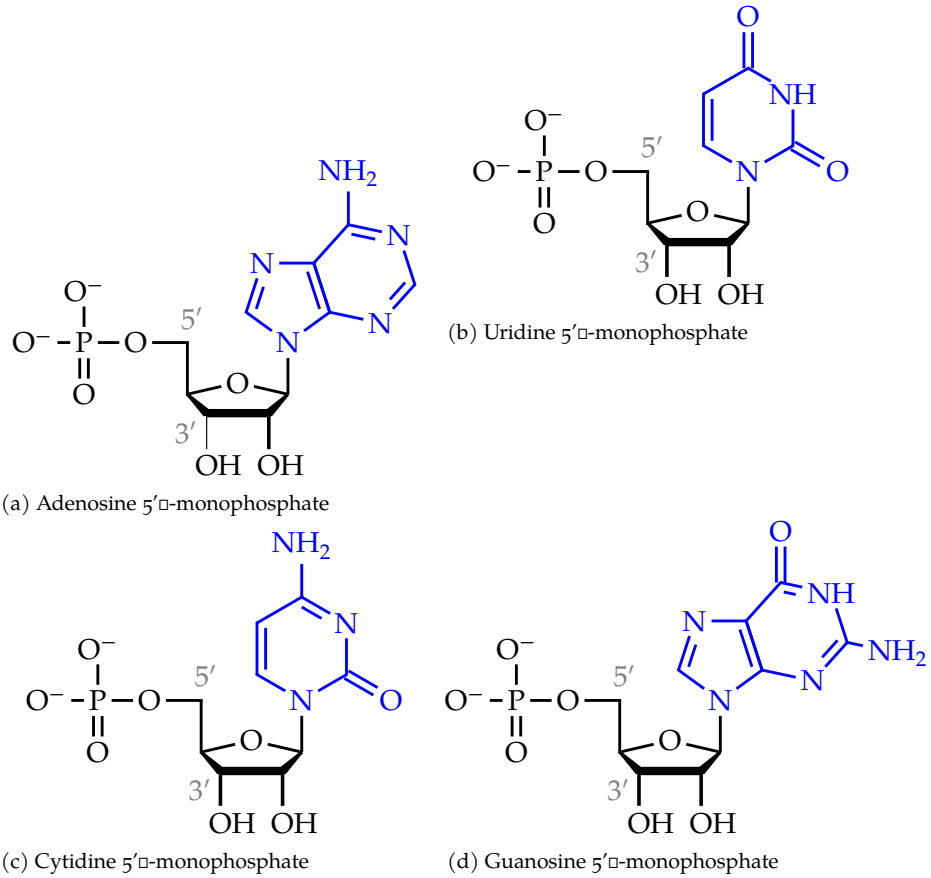
Several works revealed also that DNAs can perform some catalytic reactions, but the implication of RNAs in most of the vital chemical reactions in living systems and its broader range of chemical reactions have motivated many scientists in the last decay to study RNA molecules in more details as an independent entity. This section of our work focuses on the biochemistry of RNA structure in general, and especially on highlighting the biological importance of non-coding RNAs.



Structure of an RNA nucleotide

1.3.1 *Biochemistry of RNA molecules*

RNAs are synthesized from DNAs through a process often termed the transcription. The transcription process takes place in cell's nucleus and it is performed by RNA polymerases. Depending on the type of cells, there are many (or one) types of RNA polymerase responsible for the synthesis of a specific type of RNAs.



(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

During transcription, an RNA polymerase uses the 3′-5′ DNA template strand to synthesize a 5′-3′ RNA strand with complementary nucleotides. Similar to DNAs, nucleotides constitute the basis of RNA molecules and each nucleotide consists of a phosphate residue, a pentose sugar and a nucleobase. We also find four different nucleotides in RNA, each of them distinguished by the nucleobase they have: Adenine (A), Cytosine (C), Guanine (G) and Uracil (U) which replaces Thymine in DNA. Figure 1.1 depicts the chemical structure of each of the four different nucleobases found in RNA. Chemically, a nucleotide is a nucleoside, which has a (mono, di, trip) phosphate residue bound to its 5′-carbon atom. The common chemical structure of a nucleotide is depicted on the right side of the page. By convention, the carbon atoms of the pentose sugar in nucleotides are numbered with primes.

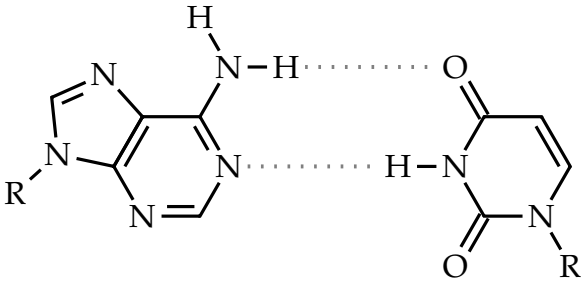
RNA molecules are simply represented as list of nucleobase characters and their functions often depend on their complex

multidimensional structures. The different nucleotides composing the RNA molecule are attached by 5′-3′ phosphatediester bonds between ribose to form the primary structure of RNA. The direction of the chain is conventionally designed as 5′ to 3′ e.i. from 5′-phosphate of the first sugar of the backbone to the 3′-hydroxyl of the last sugar in the sequence. The process in which RNA sequences are mapped to their corresponding structures is called RNA folding. In nature, RNA folding is thought to be hierarchical [2,20 from lemerlaeu GECCO]. 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 RNA structures. In contrast to the RNA primary structure, the secondary structure consists of a list of nucleobase pairs and the base pairs are formed via hydrogen bonds between the bases. 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 [**seeman1976rna**, **rosenberg1976rna**] (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 interaction.

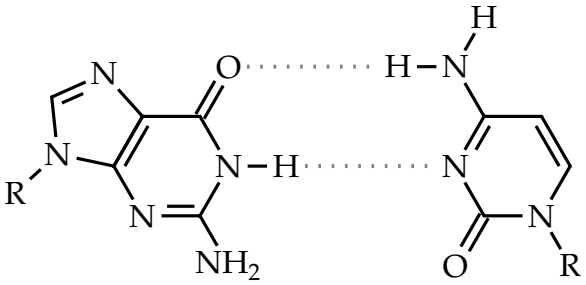
Additionally to the canonical and non-canonical pairs, we also find crossing or pseudoknotted interactions in natural RNA and they play vital roles in realising biological functions. Pseudoknots occur when two canonical or non-canonical interactions cross each other [**beyongWCpairs**]. Even though pseudoknots are often considered to be the beginning of the interaction between the secondary and tertiary levels of RNA structures, we consider them to be part of the secondary structure. Therefore, two main secondary structure definitions are considered in this work: 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 allowed.

1.3.2 *Biological function of non-coding RNAs*

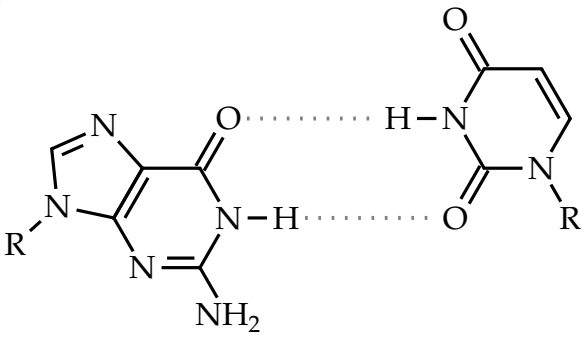
This section may be included in the section 1.1. Nevertheless, this will provide a particular introduction to some non-coding RNAs and underlay their biological significances. e.g. Aptamers & Riboswitches, SELEX, etc...



(a) Adenine-Uracil Interaction



(b) Guanine-Cytosine Interaction



(c) Guanine-Uracil Interaction

Figure 1.2: RNA base pair interactions.

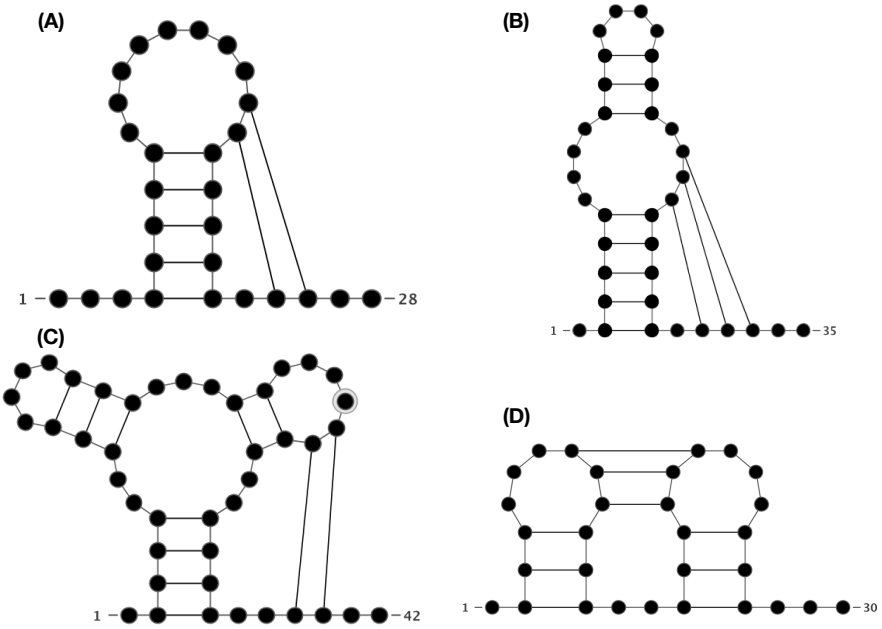


Figure 1.3: RNA pseudonotted interactions

1.4 BIOINFORMATIC DEFINITIONS AND RNA CONCEPTS.

In order to computationally study and analyse RNA molecules, a more formal representation of RNAs and bioinformatic definitions are required. We provide in this section, formal definitions and concepts that will support the result presented in this thesis.

DEFINITION 1 (RNA sequence): Let S be an RNA sequence of length L. More formally, S consists of an ordered sequence of nucleotides that can be represented as:

$$S = (S_1, ..., S_L) \text{ where } S_i \in \{A, C, G, U\}$$

S is often known as the primary structure of RNA.

DEFINITION 2 (RNA pseudoknot-free secondary structure): Given an RNA sequence $S \in \{A, C, G, U\}^L$, let $P = \{(i, j) : i < j\}$ be the list of possible pairing positions over the sequence S. A pseudoknot-free secondary structure $P_\sigma \subset P$ of such sequence S is a list of base pairs with the following constraints:

1. A nucleotide (sequence position) can only belong to a single pair, i.e. $\forall (i, j), (k, l) \in P_\sigma$ with $i < k$: $i = k \Rightarrow j = l$.
2. Paired bases must be separated by at least three bases. i.e. $\forall (i, j) \in \sigma \Rightarrow j - i > 3$.

3. There are no pseudoknots, i.e. $\nexists\,(i,j)\,,(k,l) \in P_\sigma$ with $i < k < j < l$,

4. The base pairs consist almost exclusively of Watson–Crick (C–G and A–U) pairs and Wobble (G–U) pairs. i.e. $\forall\ (i,j) \in P_\sigma \Rightarrow S_iS_j \in \{GC, CG, AU, UA, GU, UG\}$,

DEFINITION 3 (Secondary structure representation): A graphical way of representing an RNA secondary structure. Let P_σ be a secondary structure of an RNA sequence S of length L. There are several representations of P_σ.

- Dot-bracket (Or string) representation: In a this representation, the secondary structure P_σ is complactly 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 $\forall (i,j) \in P_\sigma$ 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 P_σ.

- Planar representation: it is the common way of representing an RNA secondary structure in which σ is presented as a graph with each vertex representing a nucleotide and an edge connecting consecutive nucleotides and base pairs.

- Circular (or circle) representation: similar to planar representation, P_σ is represented as 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.

- Linear representation: In this representation, P_σ is a graph in which the nucleotides are arrange consecutively in a line and the edges representing base pairs form semi-circle that do not intersect for pseudoknot-free structure.

- Mountain representation: it is mostly used for representing large structures.P_σ is presented in a two-dimensional graph, in which the x-coordinate is the position k of the nucleotide in the sequence S and the y-coordinate the number $m(k)$ of base pairs that enclose nucleotide k.

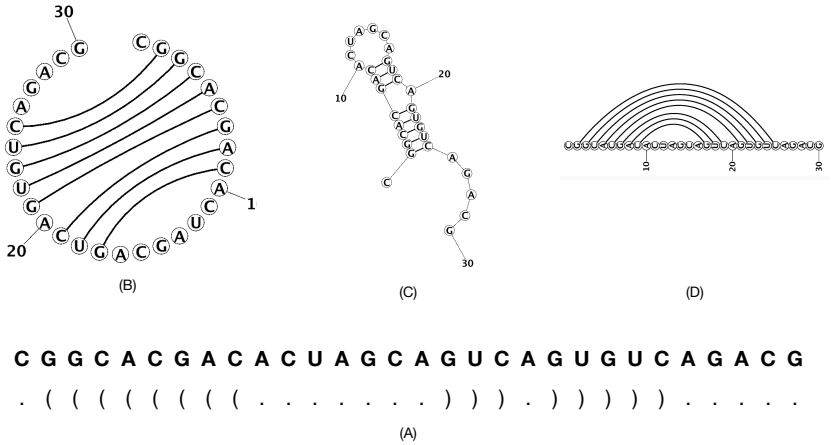


Figure 1.4: RNA secondary structure representation

- Tree representation: P_σ 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 P_σ by single matching brackets and the components are labelled with H(Hairpin), B(Bulge), I (interior loop),M (multi-loop) and S (stacking loop).

Figure 1.4 shows some examples of RNA secondary structure representation.

DEFINITION 4 (Secondary structure loop): Given a secondary structure σ over an RNA sequence S of length L, there exists a unique decomposition of σ into a set of loops $\mathcal{L}_\sigma$, where loops are the faces of its planar drawing. Each loop is characterised by its length l (the number of unpaired nucleotides in the loop) and its degree k (the number of base pairs delimiting the loop, including the closing loop pair). Therefore, $\forall \phi \in \mathcal{L}_\sigma \Rightarrow \phi = \phi_p \cup \phi_u$ where ϕ_p and ϕ_u denote respectively the set of loop base pairs and the unpaired positions. ϕ_p contains the closing loop and the interior loop pairs. We say $(i,j) \in \phi_p$ is a closing pair if and only if $\forall \phi_p \ni (i',j') \neq (i,j) : i < i' < j' < j$.

1. Interior loop: a loop with degree $k = 2$ i.e $|\phi_p| = 2$ and $\phi_u \in \{P_\sigma \cup \emptyset\}$.

2. Stacking pair: an interior loop of length $l = 0$ i.e. $\phi_p = \{(i_1, j_1), (i_2, j_2)\}$ where, (i_1, j_1) is a closing pair and (i_2, j_2) is an interior pair and, $\phi_u = \emptyset$.

3. Hairpin Loop: Any loop of degree $k = 1$. i.e $\phi_p = \{(i_1, j_1)\} = \phi_c$ and $\phi_o \neq \emptyset$.

4. Bulge loop: a special case of interior loop in which there are unpaired bases only on one side. i.e $\phi_p = \{(i, j), (l, k)\}$ with $i \neq k, j \neq l$ one of the following assumption holds:
   - If $\exists i' \in \phi_u | i < i' < j \Rightarrow \nexists k' \in \phi_u | k < k' < l$
   - If $\exists k' \in \phi_u | k < k' < l \Rightarrow \nexists i' \in \phi_u | i < i' < j$

5. Multi-loop: Any loop with degree $k > 2$ i.e. $\phi_p = \{(i_1, j_1), (i_2, j_2), \dots, (i_k, j_k)\}$ and $\phi_u \in \{P_\sigma \cup \emptyset\}$.

6. Exterior loop: a loop in which all the positions are not interior of any pair i.e. $\phi_c = \emptyset$ and $\phi_o \neq \emptyset$.

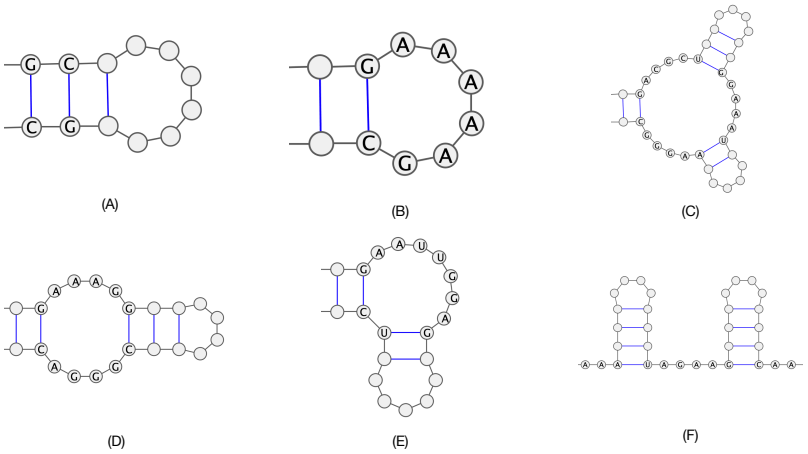


Figure 1.5: RNA secondary structure loop decomposition

DEFINITION 5 (Free Energy of an RNA secondary structure): it defines the thermodynamic stability of the secondary structure σ and it is denoted ΔG_σ. ΔG_σ is the free energy difference with respect to the completely unfolded state. The free energy of a secondary structure is computed using the According to the additivity principle [**dill97_addit_princ_bioch**], the free energy of a structure can be approximated by the sum of its constituent loops free energies. Many models allow to compute the free energies

of those constituent loops, but the dominant one is the nearest-neighbor loop energy model [**turner09_nndb**]. This model associates tabulated free energy values to loop types and nucleotide compositions; the Turner2004 [**mathews2004incorporating**] is one of the most widely used parameter sets. This structure decomposition allows an efficient dynamic programming algorithm that can determine the minimum free energy (MFE) structure of a sequence in the entire structure space. The gold standard for free-energy-based predictions is usually the MFE; however, it represents one structural estimate among many others, such as the maximum expected accuracy (MEA).

Definition 4 (Structure Ensemble):

Definition 3 (MFE secondary structure): To predict biologically relevant structures, most computational methods search for structures that minimize this free energy.

Definition 5 (Secondary structure probability)

Definition 6 (Base pair probability):

Definition 7 (Partition function of RNA) :

Definition 8 (Base pair probability matrix):

Definition 9 (Neutral set of RNA sequences) :

Definition 10 (Neutral Network):

Definition 11 (PPV) :

Definition 12 (FPV) :

Definition 13 (FFT) :

Definition 14 (Hamming Distance between two SS):

Definition 15 Ensemble defect (ED) [**zadeh2011nucleic**]: 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 as follows:

$$\begin{aligned} ED(\phi, \sigma*) &= \sum_{\sigma \in \Gamma} p(\phi, \sigma) d(\sigma, \sigma*) \\ &= L - \sum_{1<i,j<L} P_{i,j}(\phi) S_{i,j}(\sigma*) \end{aligned} \tag{1.1}$$

where $P_{i,j}$ is the base pair probability matrix and $S(s)$ is the structure matrix with entries $S_{i,j} \in \{0, 1\}$. If the structure $s$contains pair $\{i, j\}$, then $S_{i,j}(s) = 1$ otherwise $S_{i,j}(s) = 0$.

Definition 16 Normalized Energy Distance (NED): the difference between the energy of a given sequence ϕ evaluated to fold into a target structure $\sigma*$ and the minimum free energy of the

sequence in its structural ensemble Γ. The value is normalized over all the sequences in a given population P.

$$NED(\phi, \sigma*) = [1 - \Delta E_{norm}(\phi, \sigma*)]^p \ \forall p > 1 \tag{1.2}$$

where,

$$\Delta E_{norm}(\phi, \sigma*) = \frac{\Delta E(\phi, \sigma*)}{\sum_{\phi \in P} \Delta E(\phi, \sigma*)} \tag{1.3}$$

and,

$$\Delta E(\phi, \sigma*) = E(\phi, \sigma*) - \arg\min_{s \in \Gamma} E(\phi, s) \tag{1.4}$$

Definition 17 (Fitness landscape) :
Definition 18 (Local minima):
Definition 19 (Global minima):
Definition 20 (Lévy Flights):
Definition 21 (Local search):

Part I

RNA FOLDING

You can put some informational part preamble text here.

2

RNA FOLDING

TODO: Provide here a short intro for the chapter

2.1 FROM RNA SEQUENCES TO RNA STRUCTURES

This section will be dedicated at describing the folding hierarchy of RNAs.

- Different ways of representing an RNA secondary structure.
- Provide an explanation on how the RNA sequences and secondary structures are related together. Neutrality by number of structures that can take a given sequence.

Examples: Another interesting measure in this context is the number of different RNA sequences that can fold into the a given secondary structure. This however requires some knowledge about the inverse folding problem which will be discussed in the next chapter....

2.2 STABILITY AND PREDICTION OF RNA SECONDARY STRUCTURES

Note: The content of this chapter is just some dummy text. It is not a real language.

State different ways to predict a stable secondary structure for a given target. e.g. computationally (In *silico*) and experimentally (In *vitro*)

- The decomposition of RNA secondary structure into loops
- Explain the thermodynamic stability of a secondary structure and how it's computed.
- State how the MFE structure is computed: (partition function (S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure.; Probability etc.. Nussinov etc...),)

2.3 ENERGY LANDSCAPE AND RNA KINETICS

This section aim to defining what the RNA energy landscape is, and different ways of representing it.

Note: For example a pic of the energy landscape of the bistable RNA.

The second part will be for:

- the partition mechanism
- the kinetics of RNA
- a short state of art on the kinetic tools.

2.4 A LITERATURE REVIEW OF RNA FOLDING TOOLS.

This section describes in details some of the existing RNA folding tools (Static folding and dynamics):

2.4.1 *Exact MFE prediction methods*

- `RNAfold`
- `ContraFold`
- `RNAStructure`
- `LinearFold`
- `pKiss`
- `RNAExplorer`
- etc..

2.4.2 *Statical methods*

- `Mxfold`
- `ContextFold`
- etc...

2.4.3 *Heuristic methods*

- `IPknot`
- `Hotknots`

2.5 CONCLUSION

Here will be a short conclusion of the chapter:

3

RAFFT: EFFICIENT PREDICTION OF FAST-FOLDING PATHWAYS OF RNAS

In this chapter, we propose a novel heuristic to predict RNA secondary structures. 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; this is in contrast 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 $\leq$ 200 nucleotides, matching the performance of recent deep-learning-based structure prediction methods. Our algorithm 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. For the CFSE, an ensemble of 68 distinct structures computed by RAFFT allowed us to produce complete folding kinetic trajectories, whereas 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. Thanks to he fast Fourier transform on which RAFFT is based, these computations are efficient, with complexity $O(L^2 \log L)$.

3.1 MATERIAL AND METHODS

3.1.1 *Folding algorithm*

We now describe the folding algorithm starting from a sequence of nucleotides $S = (S_1 \dots S_L)$ of length L, and its associated unfolded structure. We first create a numerical representation of S

where each nucleotide is replaced by a unit vector of 4 components:

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

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

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

For example, $X^A(i) = 1$ if $S_i = A$. Next, we create a second copy $\bar{S} = (\bar{S_L} \dots \bar{S_1})$ for which we reversed the sequence order. Then, each nucleotide of $\bar{S}$ is replaced by one of the following unit vectors:

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

$\bar{A}$ (respectively $\bar{U}, \bar{C}, \bar{G}$) 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 $\text{cor}(k)$. This correlation is defined as the sum of individual X and $\bar{X}$ row correlations:

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

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

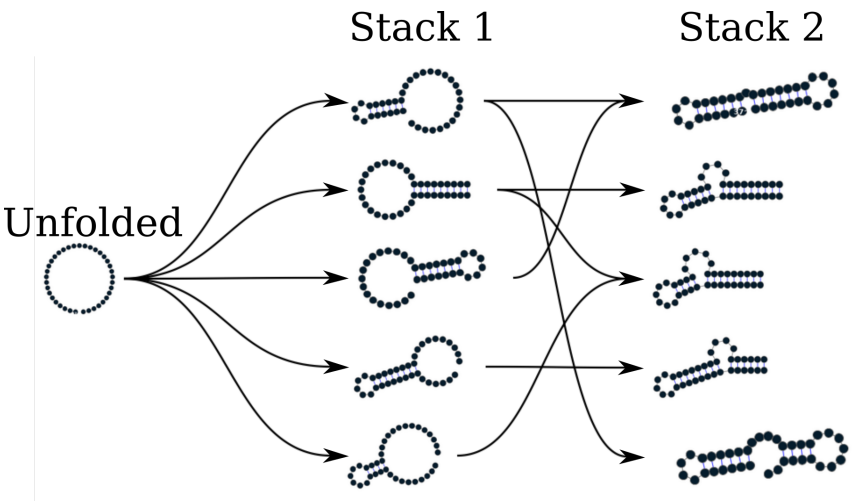


Figure 3.1: **Fast folding graph constructed using RAFFT.** In this example, the sequence is folded in two steps. The algorithm starts with the unfolded structure on the left. The $N = 5$ best stems are stored in stack 1. From stack 1, multiple stems formation are considered, but only the $N = 5$ best 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 [6].

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 [14].

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,

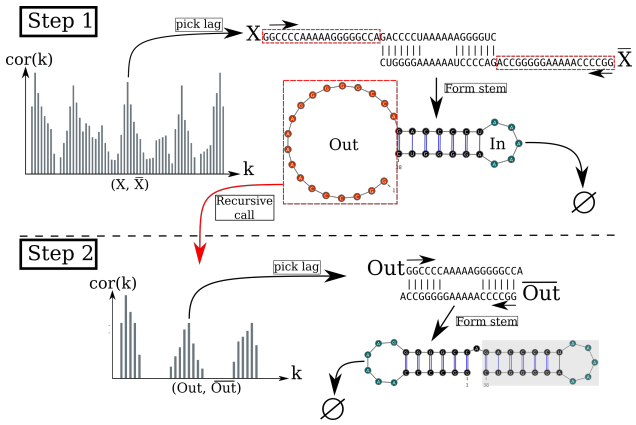


Figure 3.2: **Algorithm execution for one example sequence which requires two steps.** (Step 1) From the correlation $cor(k)$, we select one peak which corresponds to a position lag k. Then, we search for the largest stem and form it. Two fragments, "In" (the interior part of the stem) and "Out" (the 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.

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 3.2 shows an example of execution to illustrate the procedure.

The complexity of this 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 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 (exactly one base pair peer stems) can be formed. The approximate complexity therefore depends on $\sum_{i=0}^{L/2}(L-2i)\log(L-2i) = O(L^2 \log L)$. Figure **??** plots the execution time of a naive implementation of `RAFFT` and that of `RNAfold` for 20 random sequences of various lengths, showing a substantial speed-up for larger sequences.

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. Figure 3.1 illustrates this modified procedure. Like the initial version, the algorithm starts with the unfolded structure; then, the $N = 5$ 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 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.

3.1.2 *Kinetic ansatz*

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 3.1, 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 [18].

As usually done, the kinetics is modelled as a continuous-time Markov chain [15], where populations of structure evolve according to transition rates. In this context, an Arrhenius formulation is commonly used to derive transition rates $r(x \to y) \propto \exp(-\beta E^{\ddagger})$, where $E^{\ddagger}$ is the activation energy separating x from y, and β is the inverse thermal energy (mol/kcal). In contrast, our kinetic ansatz uses transition rates $r(x \to y)$ based on the Metropolis scheme already used in [12], and defined as

$$r(x \to y) = k_0 \times \min(1, \exp(-\beta \Delta\Delta G(x \to y))), \tag{3.6}$$

where $\Delta\Delta G(x \to y)$ is the stability change between structure x and y. Here k_0 is a conversion constant that we set to 1 for the sake

of simplicity. These transitions are only allowed if y is connected to x in the graph (i.e. y is in the neighborhood of x, $y \in \mathcal{X}$). Here, we initialize the population $p_x(0)$ with only unfolded structures; therefore, the trajectory represents a complete folding process. The frequency of a structure x evolves according to the master equation

$$\frac{\mathrm{d}p_x(t)}{\mathrm{d}t} = \sum_{y \in \mathcal{X}} r(y \to x)p_y(t) - r(x \to y)p_x(t), \tag{3.7}$$

where the sum runs over the neighborhood $\mathcal{X}$ of x.

The traditional kinetic approach 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 `barriers`. Then, following the Arrhenius formulation, one simulates a coarse grained kinetics between basins. In contrast, the Metropolis scheme used in our kinetic ansatz is based on the stability difference between structures, which may hide energy barriers. Due to this approximation, we referred to our approach as a 'kinetic ansatz'.

3.1.3 *Benchmark dataset used and structure prediction protocol.*

To build the dataset for the folding task application, we started from the `ArchiveII` dataset derived from multiple sources [2–5, 7, 9–11, 17, 20, 21, 24–26, 28–30]. 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 $\Delta G_s > 0$. 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.

To evaluate the structure prediction accuracy of the proposed method, we compared it to two structure estimates: the MFE structure and the ML structure. To compute the MFE structure, we used `RNAfold 2.4.13` with the default parameters and the Turner2004 set of energy parameters. We computed the prediction using `Mxfold2 0.1.1` with the default parameters for the ML

structure. Therefore, only one structure prediction per sequence for those two methods 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 search of low energy structure only. 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 above, 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 positive predictive value (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 as follows:

$$PPV = \frac{TP}{TP + FP}, \quad \text{Sensitivity} = \frac{TP}{TP + FN}, \tag{3.8}$$

where TP, FN, and FP stand respectively for the number of correctly predicted base pairs (true positives), the number of base pairs not detected (false negatives), and the number of wrongly predicted base pairs (false positives). To be consistent with previous studies, we computed these metrics using the `scorer` tool provided by Matthews *et al.* [16], which also provides a more flexible estimate where shifts are allowed.

3.1.4 *Structure space visualization*

We used a Principal Component Analysis (PCA) to visualize the loop diversity in the datasets 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 [22]. 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 (PC). Figures 3.3C and 3.3D show the two principal components of the benchmark dataset, the predicted structure using `RAFFT`, `RNAfold` (MFE-structure) and `MxFold2` (ML-structure), where the arrows represent the direction of each feature in the PC space.

3.2 EXPERIMENTAL RESULTS

3.2.1 *Application to the folding task*

We started by analyzing the prediction performances with respect to sequence lengths: we averaged the performances at fixed sequence length. Figure 3.3A shows the performance in predicted positive values (PPV) and sensitivity for the three methods. It shows that the ML method consistently outperformed `RAFFT` and MFE predictions. The t-test between the ML and the MFE predictions 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 the MFE predictions; however, `RAFFT` is significantly less accurate (p-value ≈ 0.0002), with a drastic loss of performance for sequences of length greater than 300 nucleotides. In contrast, when only the most accurate predicted structure among the 50 recorded structures per sequence was considered, we obtained 57.9% of PPV and 63.2% of sensitivity on average. The gain of performance was even more substantial for sequences of length below 200 nucleotides. The PPV was 79.4%, and the sensitivity was 81.2%. In contrast, longer sequences did not display any gain. The average performances 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].

All methods performed poorly on two groups of sequences: one group of 80 nucleotides long RNAs, and the second group of around 200 nucleotides. Figure 3.3B shows three examples of such sequences. Both groups have large unpaired regions, which for the first group lead to structures with average free energies

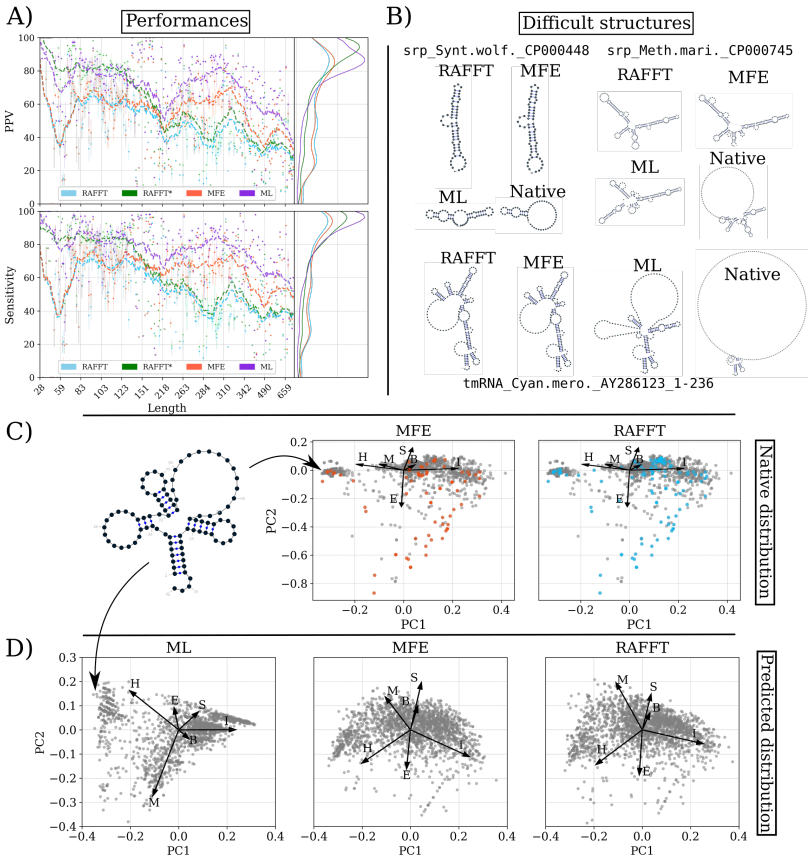


Figure 3.3: **RAFFT's performance on folding task.** (A) PPV and sensitivity *vs* sequence length. In the left panels, RAFFT (in blue) shows the scores when for the structure (out of $N = 50$ predictions) with the lowest free energy, whereas RAFFT* (in green) shows the best PPV score in that ensemble. Each dot corresponds to the mean performance for a given sequence length, and vertical lines display their standard deviation. The right panels of both figures show the distribution of PPV and sensitivity sequence-wise. (B) List of structures that are challenging to predict using the thermodynamic model. The sequence names are from the dataset. The native structures have large regions with unpaired nucleotides. (C) PCA for structures in the dataset. An example of structures with a large hairpin (**H**) is shown on the left. The points marked in orange (resp. blue) are the MFE (resp. RAFFT) structures with PPV $\leq 10\%$. (D) PCA for the predicted structures. The MFE and RAFFT structure spaces look similar and more diverse than the ML structure space, closer to the native structure space.

9.8 kcal/mol according to our dataset. The PCA analysis of the native structure space, shown in Figure 3.3C, reveals a propensity

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. For the ML and MFE only one prediction per sequence and for RAFFT 50 predictions per sequence were used. Here RAFFT (respectively RAFFT*) refers to the case when the lowest free energy (resp. highest PPV) out of the 50 predictions is selected.

	RAFFT	RAFFT*	MFE	ML
	All sequences			
PPV	47.7	60.0	55.9	70.4
Sensitivity	52.8	62.8	63.3	77.1
	Sequences with lengths ≤ 200			
PPV	57.9	79.4	59.5	76.7
Sensitivity	63.2	81.2	65.5	82.9

for interior loops and the presence of large unpaired regions like hairpins or external loops. Figure 3.3D shows the structure space produced by the ML predictions, which seems close to the native structure space. In contrast, the structure spaces produced by RAFFT and RNAfold (MFE) are similar and more diverse.

3.2.2 *Selected applications of the kinetic ansatz*

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 pseudoknot which is not taken into account here.

Figures 3.4A and 3.4B show respectively the fast-folding graph constructed using RAFFT, and the MFE and native structures 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

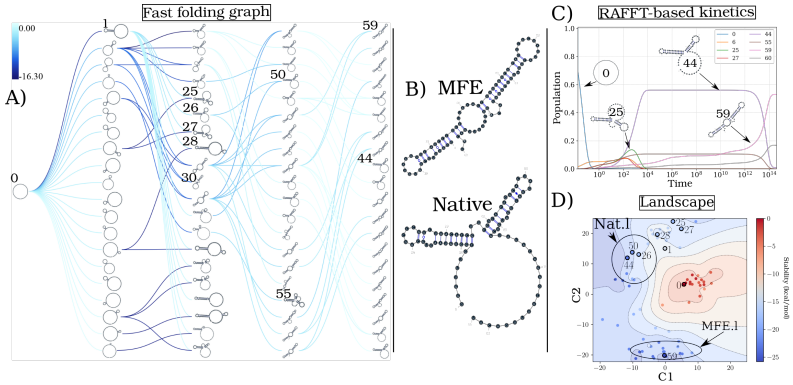


Figure 3.4: **Application of the folding kinetic ansatz on CFSE.** (A) Fast-folding graph in four steps and $N = 20$ structures stored in a stack at each step. The edges are coloured according to $\Delta\Delta G$. At each step, the structures are ordered by their free energy from top to bottom. The minimum 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.

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 3.4D shows the landscape interpolated with all the structures found; this landscape illustrates the bi-stability of the CFSE, where the native and MFE structures are in distinct regions of the structure space.

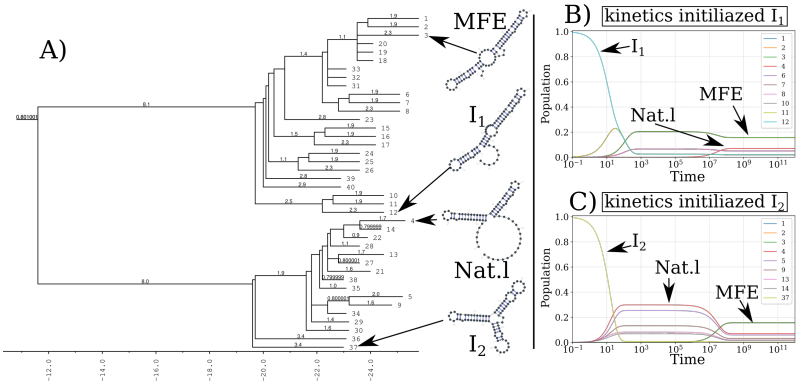


Figure 3.5: **Folding kinetics of CFSE using `Treekin`.** A) Barrier tree of the CFSE. From a set of 1.5×10^6 sub-optimal structures, 40 local minima were found, connected through saddle points. The tree shows two alternative structures separated by a high barrier with the global minimum (MFE structure) 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.

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 Eq 3.6. Starting from a population of unfolded structure and using the computed transition rates, the native of structures is calculated using Eq 3.7. Figure 3.4C 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 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` [8]. First, we generated 1.5×10^6 sub-optimal structures up to 15 kcal/mol above the MFE structure using `RNAsubopt` [14]. 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` [8], with the connectivity between basins represented as a barrier tree (see Figure 3.5A). 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 3.5B and 3.5C, respectively). In Figure 3.5B, 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 `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 `GGCCCCUUUGGGGGCCAGACCCCUAAAGGGGUC`. For `Treekin`, we first sampled the whole space of 20×10^3 suboptimal 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 3.6 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 3.6A and 3.6B. Secondly, we simulate the kinetics starting from the two initial conditions (See Figure 3.6B). 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 3.6B' 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.

3.3 CONCLUSION

We have proposed a method for RNA structure and dynamics predictions called `RAFFT`. Our method was inspired by the experimental observation of parallel fast-folding pathways. We designed an algorithm that produces parallel folding pathways, in which stems are formed sequentially, to mimic this observation.

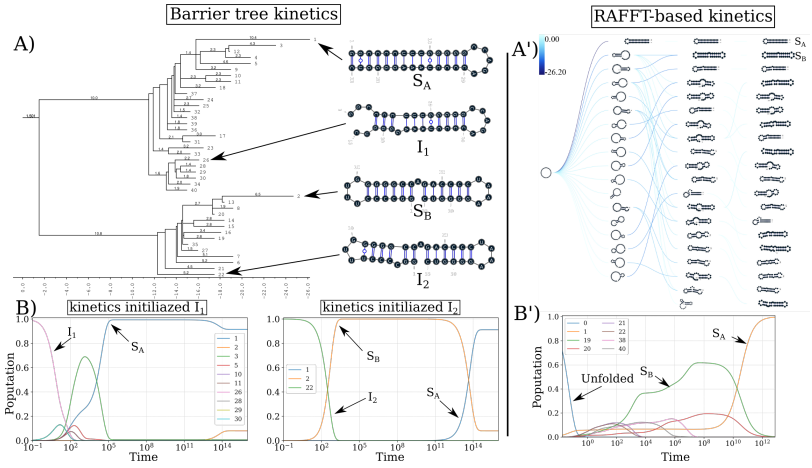


Figure 3.6: **RAFFT *vs* Treekin: folding kinetics of a bi-stable RNA sequence.** (A) Barrier tree for the bi-stable example sequence. The local minima and the corresponding barriers are computed from the complete enumeration of the structure space. The bi-stability is visible on the barrier tree through 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.

Then, to model the slow part of the folding process, we proposed a kinetic ansatz that exploits the parallel fast-folding pathways predicted.

First, we compared the algorithm performance for the folding task. Two structure estimates were compared with our method: the MFE structure computed using RNAfold, and the ML estimate using MxFold2. Other thermodynamic-based and ML-based tools were investigated but not shown here because their performances were found to be very similar to the one of MxFold2 and RNAfold (See SI for the complete benchmark). When we considered the lowest energy structure, the comparison of RAFFT to existing tools confirmed the overall validity of our approach. In more detail,

comparison with thermodynamic/ML models yielded the following results. First, the ML predictions performed consistently better than both RAFFT and the MFE 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 correlate with the ones in the benchmark dataset, which a PCA analysis revealed too. In contrast, the density of loops was lower in the structure spaces produced by RAFFT and MFE, implying some over-fitting in the ML model. Finally, known structures obtained through covariation analysis reflect structures *in vivo* conditions. Therefore, the structures predicted by ML methods may not only result from their sequences alone but also from their molecular environment, e.g. chaperones. We expect the thermodynamic methods to provide a more robust framework for the study of sequence-to-structure relations.

With respect to MFE-based tools, we obtained a substantial gain of performance when analyzing $N = 50$ predicted structures per sequence and 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 produce better structures, 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, albeit using many fewer structures.

However, further improvements and extensions of the algorithm may be investigated. For starters, the choice of stems is limited to the largest in each positional lag, a greedy choice which may not be optimal. Furthermore, we have constructed parallel pathways leading to a diversity of accessible structures, but 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 the identification of long-lived intermediates and use 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.

DATA AVAILABILITY

The implementation in `python3.0` of RAFFT and the benchmark data used in this manuscript are available at `https://github.com/strevol-mpi-mis/RAFFT`. We also provide the scripts used for the figures and kinetic analyses.

Part II

RNA DESIGN

4

RNA DESIGN

TODO: Here will be a short introduction to the chapter.

4.1 BIOLOGICAL MOTIVATION AND BIOTECHNOLOGICAL IMPLICATIONS

This aim to provide a biological motivation and the biotechnological implications of the RNA design

You might get unexpected results using math in chapter or section heads. Consider the `pdfspacing` option.

4.2 POSITIVE AND NEGATIVE DESIGN.

List of tasks to do

- Define the positive and negative design.
- Highlight the difference between the two types of designs.
- Provide a formal definition of the type of design on which this thesis is focused (RNA inverse folding).

4.3 OBJECTIVE FUNCTIONS PREVIOUSLY USED.

this section will provide most of the objective functions used in the RNA design

4.4 A REVIEW ON EXISTING INVERSE RNA FOLDING TOOLS.

4.4.1 *Pseudoknot-free RNA inverse folding tools*

- `RNAinverse`
- `RNAPong`
- `SendRNA`
- `ERD` (Evolutionary RNA Design) ERD or Evolutionary RNA Design (Esmaili-Taheri 2015) is a recent program, first developed in 2014 (Esmaili-Taheri 2014) and one year after,

an updated version has been released. It starts by decomposing the target structure into structural components (generally called loops) and then, independently uses an evolutionary algorithm to minimize each corresponding subsequence energy to recombine the different fragments to form the designed sequence finally. The main lines of ERD are: Pool reconstruction: using a collection of RNA sequences (STRAN database) similar to the natural ones, a pool of sequence is constructed with respect to their length by successively finding the corresponding structure using ViennaRNA, decomposing the structure in subcomponents, and finally the corresponding sub-sequences of the same length are gathered to form a pool. 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. Sequence initialisation: after decomposing the target structure in 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. Evolutionary optimization of the sub-sequences: to improve the initial sequence, an evolutionary algorithm is performed on each sub-components, and 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 sub-sequence corresponding to a sub-component by a new random one from the pool with respect to the length, and a selection which consists of choosing from a population of 15 RNA sequences or sub-sequences, 3 best sequences with respect to their free energy and adding them to the best from the preview generation, 3 best ones with respect to the Hamming distance from the target are therefore chosen. The next-generation population is then obtained by generating for each of the three best sequences 5 new sequences.

- `MODENA` `MODENA` or Multi-objective Design of Nucleic Acids (Taneda 2011) is a multi-objective genetic algorithm that explores

the approximate set of weak Pareto optimal solution in the space of two objective functions: one that measures the structure stability and another one that measures the similarities between the predicted secondary structure of the designed sequence and the target in order to the dominant solution. More precisely, let

- NEMO [19]
- SentRNA [23]
- antaRNA
- RNAinverse
- NUPACK
- RNAiFold
- IncaRNAtion
- INFO-RNA
- Frnakenstein
- RNAfbinv
- RNA-SSD
- DSS-Opt
- LeaRNA

4.4.2 *Pseudoknotted RNA inverse folding tools*

- Inv
- antaRNA
- MODENA

4.5 CONCLUSION

Provide here a short conclusion of the chapter.

5

ARNAQUE: AN EVOLUTIONARY ALGORITHM FOR INVERSE FOLDING INSPIRED BY LÉVY FLIGHTS.

A Lévy flight is a random walk with step sizes that follow a heavy-tailed probability distribution. This type of random walk, with many small steps and a few large ones, has inspired many applications in genetic programming and evolutionary algorithms in recent years, but is yet to be applied to RNA design. Here we study the inverse folding problem for RNA, viz. the discovery of sequences that fold into given target secondary structures. We implement a Lévy mutation scheme in an updated version of `aRNAque`, an evolutionary inverse folding algorithm, and apply it to the design of RNAs with and without pseudoknots. We find that the Lévy mutation scheme increases the diversity of designed RNA sequences and reduces the average number of evaluations of the evolutionary algorithm. The results show improved performance on both `Pseudobase++` and the `Eterna100` datasets, outperforming existing inverse folding tools. We propose that a Lévy flight offers a better standard mutation scheme for optimizing RNA design.

We provide in this work an updated version of `aRNAque` supporting pseudoknotted RNA target structures. In addition to the support for pseudoknots, we provide an updated mutation mode based on a Zipf distribution. For a given population of RNA sequences, an exponent c of the Zipf distribution, and the mutation parameters: P_N and P_C, we present the mutation algorithm in Algorithm 1.

5.1 MATERIAL AND METHODS

5.1.1 *Inverse folding evolutionary algorithm (EA)*

Below, we provide a brief overview of our evolutionary search algorithm and our mutation scheme. 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:

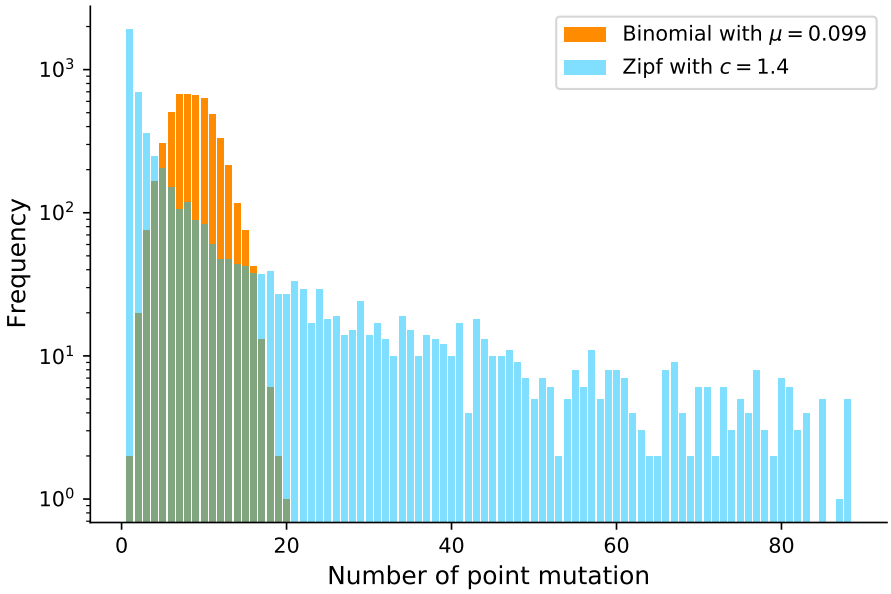


Figure 5.1: 5000 samplings Binomial and Zipf distributions. Both distributions have a mean of 8.7 point mutations for a sequence of length 88 nucleotides.

- 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 `aRNAque` is provided in [**merleau2021simple**].
- 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 subsection.

Mutation mode

For a given target RNA secondary structure $\sigma*$ of length L, the space of potential solutions to the inverse folding problem is $S = \{A, C, G, U\}^L$. An evolutionary algorithm explores the space S through its move (or mutation) operator. Given a sequence

$\phi \in S$, a sequence $\phi' \in S$ is said to be an n-point mutation of ϕ if it differs from ϕ at n nucleotides; i.e. $h(\phi, \phi') = n$ where $h(.,.)$ is the hamming distance on S.

A mutation mode is a random variable U taking values in $\{1, ..., L\}$. $P(U = n)$ 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. 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 Figure 3 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 $\mu \approx 1/L$.

5.1.2 *Parameter analysis and benchmark*

Here we analyse mutation parameters and compare local and Lévy mutation modes.

5.1.2.1 *Benchmark data used*

To compare our new version of `aRNAque` with existing tools in the literature, we used the `PseudoBase++` benchmark datasets for

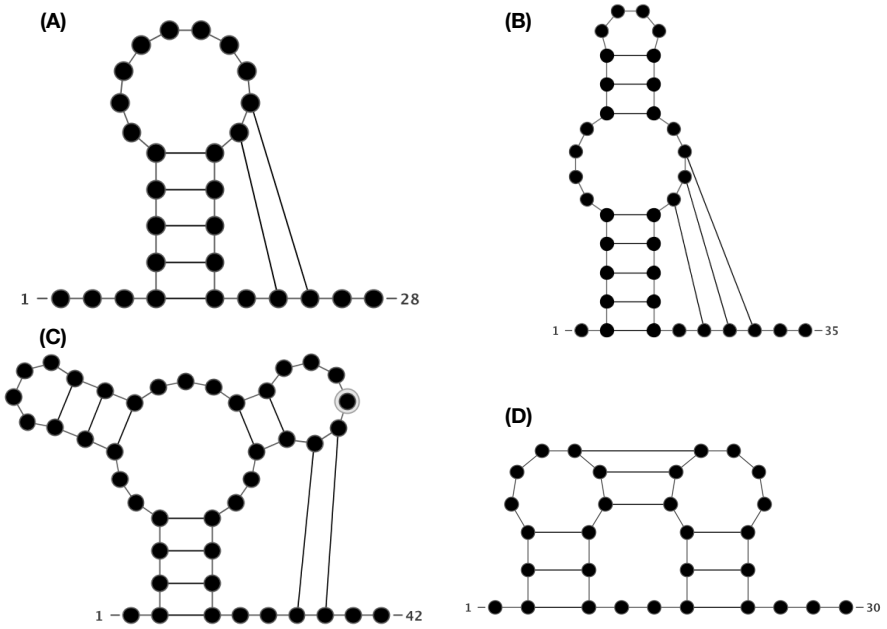


Figure 5.2: Types of pseudoknots accommodated by aRNAque. (A) Hairpin (H-type) pseudoknot. (B) Bulge (B-type) pseudoknot. (C) Complex hairpin (cH-type) pseudoknot. (D) Kissing hairpin (K-type) pseudoknot.

pseudoknotted target structures and the `Eterna100` dataset for pseudoknot-free target structures.

The `PseudoBase++` is a set of 265 pseudoknotted RNA structures used to benchmark `Modena`. It was initially 304 RNA secondary structures, but we excluded 37 because they had non-canonical base pairs. We then grouped the structures into four pseudoknot motifs (Figure 5.2): 209 hairpin pseudoknots (H), 29 bulge pseudoknots (B), 8 complex hairpin pseudoknots (cH) and 4 kissing hairpin pseudoknots (K).

The `Eterna100` dataset [**Eterna**] 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 [27]. Out of the 100 targets secondary structures, 19 turned out to be unsolvable using the recent version of `ViennaRNA` (Version 2.14). Subsequently, an `Eterna100-V2` [**Eterna**] was released in which the 19 targets were slightly modified to be solvable using `ViennaRNA 2.14`.

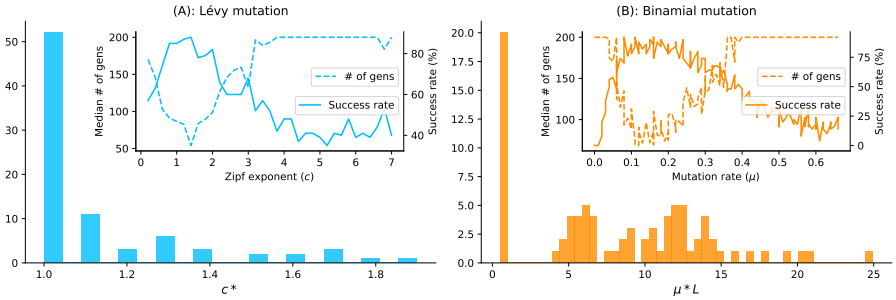


Figure 5.3: Parameter tuning for both binomial and Levy mutation schemes.(A) Lévy-flight parameter tuning. Histogram of best exponent parameter ($c*$) for a set of 81 target structures with different pseudoknot patterns and various lengths. The most frequent best exponent value is 1. Note that this corresponds to an exponent near 0 for the complementary cumulative distribution function. The inset figure shows the median generations and the success percentage *vs.* the exponent parameter (c) for one of the pseudoknotted target of length 88 with a broader range of c (from 0 to 7 with a step size of 0.1). (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 mutation rate seems to be the high one for different lengths as well. Similar to the Levy mutation, The inset figure shows the median generations and the percentage of success *vs.* the mutation rate (μ) for one of the pseudoknotte

.

5.1.3 *Benchmark protocol*

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 (265 from `PseudoBase++` and 100 from `EteRNA100`). First, for each of the 365 target structures σ^* in the datasets, 20 sequences were designed. To measure the performance of each tool, each designed sequence s is folded into a secondary structure σ and the similarities between σ and σ^* are computed using the base pair distance. Second, for each of the `Eterna100` target structures and a maximum of 5000 generations (i.e. 50,000 evaluations), 5 to 20 runs were launched independently, which results in at least 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 sequence folds into the target structure (i.e. the Hamming distance between the target structure and the MFE structure is 0).

5.1.3.1 *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 analysis presented here, we used `IPknot`, and both `HotKnots` and `IPknot` for pseudoknotted targets. Furthermore, we considered `pkiss`, a well know tool for K-type pseudoknot prediction, but since the `PseudoBase++` dataset contains just 5 K-type pseudoknotted structures and `pKiss` has higher time complexity ($O(n^6)$), we did not find it efficient for the benchmark we performed here.

5.1.3.2 *Mutation parameters tuning*

One of the main challenges for an evolutionary algorithm is to find optimum parameters such as mutation rate, population size and selection function. We used 81 pseudoknotted targets with lengths from 25 to 181 nucleotides for the mutation parameter analysis. We set the maximum number of generations to 200 and the population size to 100. The best Lévy mutation parameter $c*$ (respectively $\mu*$ for the binomial mutation) has the lowest median number of generations.

- Binomial mutation: First, for each $\mu \in [0, 1]$ with a step size of 0.005, 50 sequences were designed using `aRNAque` and the input pseudoknotted target structure was `PKB00342`. In Figure 5.3B, the inset figure shows the median number of generations and the success rate for each parameter μ. The best mutation rate is $\mu* = 0.085$ (with a median number of generation 93.5 and a success rate of 92%). 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. Second, for the 80 target structures with pseudoknots, 20 sequences were designed for $\mu \in [0, 0.2]$ with a step size of $1/L$. Figure 5.3B shows the histogram of the best mutation rate found for each target structure. Two main regimes are visible: one regime in which the best mutation rate is the low one ($\approx 1/L$) and the second regime for which for which the high mutation rate was optimal.

- Lévy mutation: we used the same dataset to tune the Zipf exponent c. First, for each $c \in [0, 7]$ with a step size of 0.1 and the same pseudoknotted target structure `PKB00342`, 100 sequences were designed using `aRNAque`. The inset of Figure 5.3A shows the median number of generations and the success rate for each exponent c respectively. The Zipf exponent distribution that produced the highest success rate and the minimum number of generations is $c* = 1.4$. Secondly, for the $c \in [1, 2]$ and a step size of 0.1, an optimum exponent parameter $c*$ was investigated for all the 80 remaining target structures. Figure 5.3A shows the histogram of $c*$. Contrary to binomial mutation, the optimum exponent parameter does not vary too much $(\forall \sigma, c* \approx 1)$.

The main observation is that when using a Lévy mutation, the optimal mutation rate is approximately independent of the target structure. In contrast, the optimum binomial mutation rate parameter $\mu*$ varies with different targets. Although both mutation modes have approximately the same success rates (88% for the Lévy over 100 runs and $\approx$ 92% for the binomial over 50 runs), the Lévy flight mutation scheme is more robust to different targets. Moreover, the median number of generations for the Lévy mutation is lower (54 for the Lévy and 92 for the binomial mutation mode), thus enhancing efficiency.

5.2 EXPERIMENTAL RESULTS

A Lévy mutation scheme offers a compromise between exploration at different scales (mostly local search combined with rare big jumps). Such a scheme significantly improves the number of evaluations needed to hit the target structure, while better avoiding getting trapped in local optima. We first compared the performance of `aRNAque` using Lévy mutations to the previous version with local mutations (binomial number of point-mutations with $\mu \approx \frac{1}{L}$). Secondly, we compared `aRNAque` to the existing pseudoknotted RNA inverse folding tool `antaRNA` using two folding tools: `HotKnots` and `IPknot`. We used the `PseudoBase++` dataset for both benchmarks.

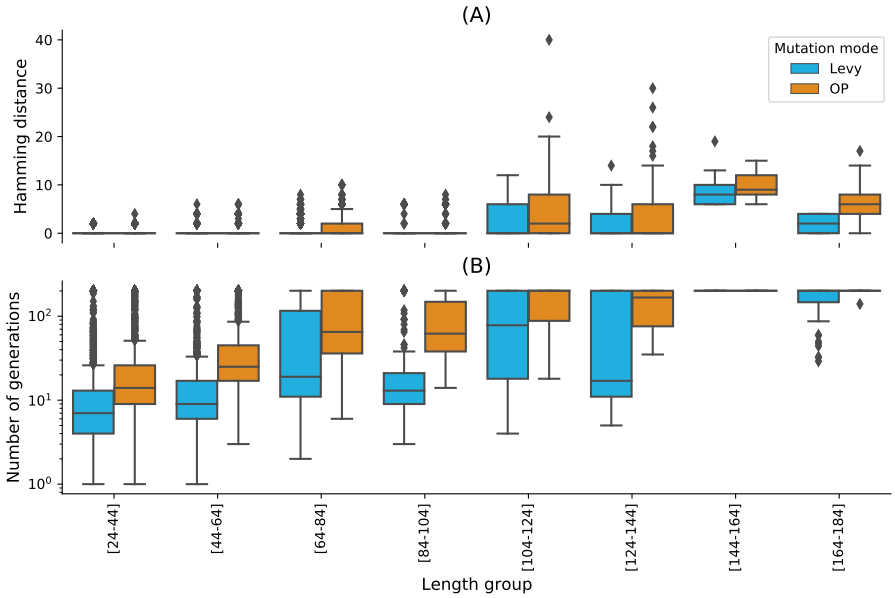


Figure 5.4: Lévy mutation mode *vs* local mutation (one-point mutation). (A) Hamming distance distributions *vs.* target structure lengths. (B) Number of generations distributions for different length groups. In both (A) and (B), lower values indicate better performance. The target structures are solvable 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.

.

5.2.1 *Performance on* `PseudoBase++`*: Levy mutation* vs. *Local mutation*

Figure 5.4 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

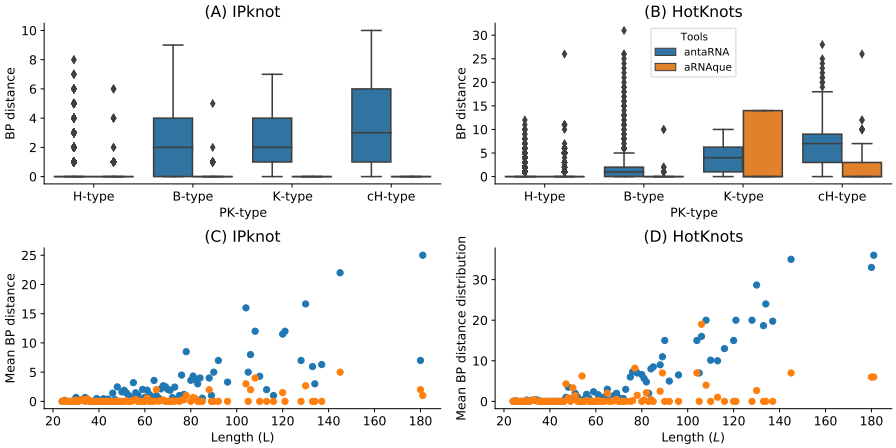


Figure 5.5: aRNAque *vs* antaRNA on PseudoBase++ dataset using both IPknot and HotKnots. (A, B) Base pair distance distributions of the designed sequences to the target structure for different pseudoknot types. (C,D) Mean base pair distance against target lengths.

.

the Lévy mutation instead of a local mutation scheme can significantly increase the performance of aRNAque. The gain was less significant 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 minimum 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 *Performance on PseudoBase++: aRNAque vs. antaRNA*

We also compared the sequences designed using aRNAque (with the Lévy mutation scheme) to those produced by antaRNA. Figures 5.5A and 5.5C 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). In sum, as target length increases, the perfor-

mance of antaRNA (local search) is considerably degraded , while aRNAque (Lévy flight search) stays almost constant.

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). Figures 5.5B and 5.5D 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 *Performance on Eterna100 dataset*

Finally, we performed a third benchmark on the Eterna100 datasets. First, on the Eterna100-V1 dataset, the Lévy flight version of aRNAque successfully designed 89% of the targets and the one-point mutation (local mutation) version achieved 91% of success. Combining the two datasets, aRNAque solved in total 92% of the targets of Eterna100-V1 (see also [**merleau2021simple**]). 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) of 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 5.7A). Second, a new benchmark was performed on Eterna100-V2 with aRNAque achieving a 93% success rate. Compared to recently reported benchmark results [**Eterna**], aRNAque achieved the 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.

5.3 CONCLUSION

In this work, we provide an updated version of `aRNAque` implementing a Lévy flight mutation scheme that supports pseudoknottted RNA secondary structures. The benefit of a Lévy flight over a purely local (binomial with $\mu << 1$ or a single point mutation) 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 and reduced the number of evaluations of the evolutionary algorithm implemented in `aRNAque`. The main advantage of using a Lévy flight over local search is a reduction in the number of generations required to reach a target. 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".

Figures 5.6C and 5.6D 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 the one 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 5.6D). Using the mean entropy of structures as an alternate measure of diversity, we see in Figure 5.6A 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 Flight over local search in target RNA structures is due to the high base

pair density of pseudoknotted structures. Given that pseudoknots present a high density of interactions, there are dramatic increases in possible incorrect folds and thus becoming trapped near local optima [**hajdin2013accurate**]. Large numbers of mutations in paired positions, as implied by a heavy tailed distribution, are necessary to explore radically different solutions.

To illustrate that Lévy Flight performance was 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 5.7B shows 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 psuedoknots

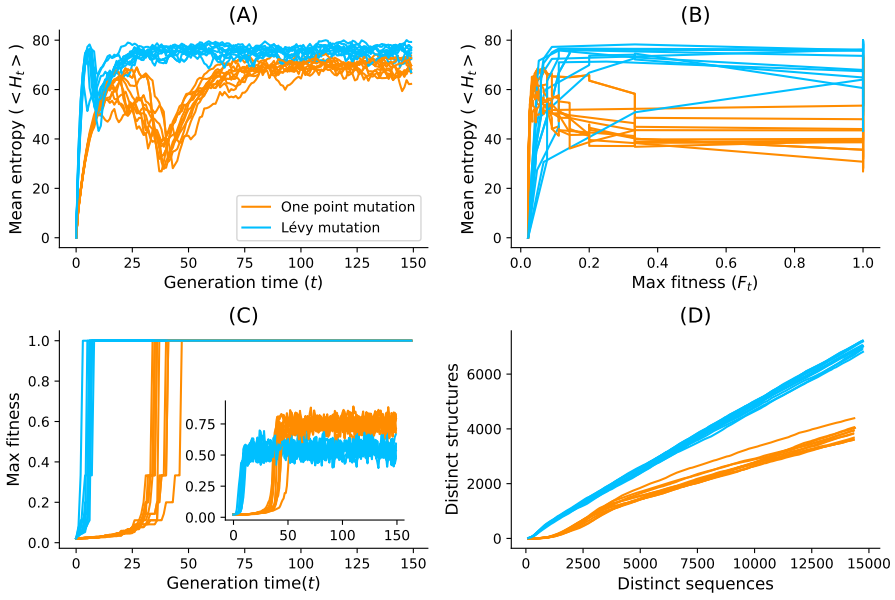


Figure 5.6: Lévy mutation *vs* one-point mutation: diversity analysis. For the `Eterna100` target structure [*CloudBeta*] *5 Adjacent Stack Multi-Branch Loop*, ten independent runs were performed in which a minimum of 10 sequences were designed per run. (A) Mean Shannon entropy of the population sequences over time for both binomial and Lévy mutation. (B) The max fitness plotted against the entropy over time. (C) Max fitness and mean fitness (inset) over time. (D) Distinct sequences *vs.* Distinct structures over time.

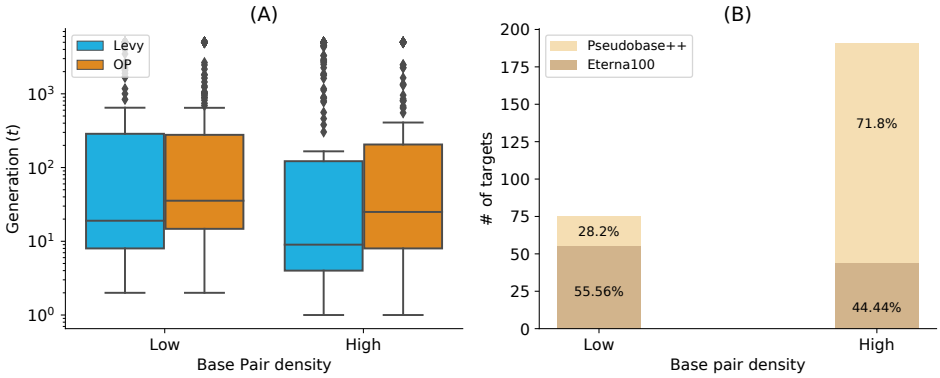


Figure 5.7: Lévy mutation *vs.* one-point mutation: performance analysis with respect to the base-pair density. The higher the base-pair density is, the more useful the Lévy mutation scheme to speed up the optimization EA. (A) Distributions of number of generations for the low and high base-pair density targets of the `Eterna100` dataset. (B) Percentages of targets with low and high base-pair density for the `Eterna100` and `PseudBase++`

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.

CONCLUSION

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. Another possible limitation is that most pseudoknotted and pseudoknot-free target structures are relatively easy to solve (in less than 100 generation time), requiring more investigations for the unsolved targets to illustrate the performance of the Lévy mutation better.

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. Not only does Lévy flight mutations lead to greater diversity of RNA se-

quence solutions, but it also reduces the evolutionary algorithm's number of evaluations, thus improving computing time.

availability

The implementation in `python3.7` of `aRNAque` and the benchmark data used in this manuscript are available at `https://github.com/strevol-mpi-mis/aRNAque`. We also provide the scripts used for the figures and the designed sequences analysis.

competing interests

The authors declare that they have no competing interests.

author's contributions

Text for this section ...

acknowledgements

We thank the Structure of Evolution Group at the Max Planck Institute for Mathematics in the Sciences, and especially Vincent Messow, for useful discussions. The Alexander von Humboldt Foundation provided funding for this work in the framework of the Sofja Kovalevskaja Award endowed by the German Federal Ministry of Education.

Algorithm 1: Mutation algorithm

/* $P' = \{S'_1 \dots S'_n\}$: the mutated population;
$P = \{S_1 \dots S_n\}$: a list of n RNA sequences to mutate;

$P_C = \{w_{AU}, w_{GU}, w_{GC}\}$: 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**
    $S' \leftarrow P_i$; // Assign the sequence $S_i \in P$ to S'
    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 S'
        $n_j \in \{A, U, C, G\} \sim P_N$; // select a random nucleotide n_j with respect to P_N
        $S'_r \leftarrow n_j$; // replace the nucleotide at position j in the RNA sequence S' with n_j
    for $j \in \{1, 2, \dots B_i\}$ **do**
        $k_j \in \{AU, UA, CG, GC, GU, UG\} \sim P_C$; // 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
        $S'_b \leftarrow k_j$; // replace respectively the nucleotides at the base pair position $b_i \in b$ by $k \in k_j$
    $P' \leftarrow P' \cup S'$; // Add S' to the list P'

Part III

DISCUSSION AND PERSPECTIVE

6

RAFFT AND CONTINUOUS TRANSITION IN EVOLUTION

Part IV

APPENDIX

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, **pages** 1–11. DOI: 10.1007/978-3-319-02624-4_1. URL: https://doi.org/10.1007/978-3-319-02624-4_1.

[2] 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), **pages** 1–10.

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

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

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

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

[7] 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), **pages** 2462–2464.

[8] 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.

[9] 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 **andothers**. "Rfam: updates to the RNA families database". **in**: *Nucleic Acids Research* 37.suppl_1 (2009), **pages** D136–D140.

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

[11] 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), **page** 3055.

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

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

[14] 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), **page** 26. DOI: 10.1186/1748-7188-6-26. URL: https://doi.org/10.1186/1748-7188-6-26.

[15] 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, **pages** 23–31. DOI: 10.5220/0008916600230031. URL: https://doi.org/10.5220/0008916600230031.

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

[17] 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), **pages** 911–940.

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

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

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

[21] 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), **pages** 701–719.

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

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

[24] 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), **pages** 1808–1818.

[25] 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), **pages** 96–97. DOI: 10.1093/nar/25.1.96. URL: https://doi.org/10.1093/nar/25.1.96.

[26] 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), **pages** 148–153.

[27] 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), **pages** D280–D282.

[28] 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), **pages** 277–291.

[29] C. Zwieb. "Tmrdb (tmRNA database)". **in**: *Nucleic Acids Research* 28.1 (2000), **pages** 169–170. DOI: 10.1093/nar/28.1.169. URL: https://doi.org/10.1093/nar/28.1.169.

[30] C. Zwieb. "Tmrdb (tmRNA database)". **in**: *Nucleic Acids Research* 31.1 (2003), **pages** 446–447. DOI: 10.1093/nar/gkg019. URL: https://doi.org/10.1093/nar/gkg019.

DECLARATION

Put your declaration here.

Saarbrücken, June 2018

André Miede & Ivo Pletikosić

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.