

1 From RNA folding to inverse folding:
2 *a computational study*

3 Der Fakultät für Mathematik und Informatik
4 der Universität Leipzig
5 eingereichte

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

7 zur Erlangung des akademischen Grades
8 DOCTOR RERUM NATURALIUM
9 (Dr.rer.nat.)

10 im Fachgebiet
11 Informatik
12 vorgelegt

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

15 Leipzig, den December 2021

[June 17, 2022 at 17:55 – 1.0]

¹⁶

Ohana means family.

¹⁷

Family means nobody gets left behind, or forgotten.

¹⁸

— Lilo & Stitch

¹⁹

Dedicated to my loving dad Micheal Saha.

[June 17, 2022 at 17:55 – 1.0]

²⁰ ABSTRACT

²¹ The understanding of biological processes at the molecular level
²² has grown immensely since the discovery of structural conforma-
²³ tions of DNA in the early 1953s by Watson and Crick. Later, they
²⁴ formulated the central dogma of molecular biology that describes
²⁵ the flow of information between DNA, RNA, and protein.

²⁶ On the one hand, the dogmatic statement that proteins are
²⁷ the only entities that can perform enzymatic functions within
²⁸ organisms has undergone a significant revision in the last few
²⁹ decades. More recent studies also revealed that DNAs could per-
³⁰ form some catalytic reactions. In addition to proteins, specific
³¹ RNA molecules, namely the non-coding RNAs, have accounted
³² for their implications in most vital chemical reactions in living
³³ systems so far exclusive to proteins.

³⁴ On the other hand, advancements in developing sophisticated
³⁵ techniques for sequencing data and intensive lab experiments
³⁶ have led to identifying more non-coding RNAs involved in realiz-
³⁷ ing many essential biological functions and their implications in
³⁸ many diseases. The advancements in studying RNA molecules
³⁹ and the current COVID-19 pandemic situation have contributed
⁴⁰ more to shifting the attention from DNA and protein to RNAs.
⁴¹ Many studies revealed that the non-coding RNA functions are
⁴² performed by high-level structures that often depend on their
⁴³ low-level structures, such as the secondary structure. This thesis
⁴⁴ studies the computational folding mechanism and inverse folding
⁴⁵ of non-coding RNAs at the secondary level.

⁴⁶ Computationally, folding an RNA molecule to its secondary
⁴⁷ structure involves finding the one with the minimum free en-
⁴⁸ ergy in the space of all possible secondary structures. In contrast,
⁴⁹ the inverse problem consists of searching in the space of RNA
⁵⁰ sequences for those whose minimum free energy structures are
⁵¹ similar to a given target structure. Addressing both problems
⁵² often requires an energy function that allows mapping each RNA
⁵³ molecule and a probable secondary structure to a free energy
⁵⁴ value. Such an energy function often relies on thermodynamic
⁵⁵ parameters experimentally measured. In this thesis, our contribu-
⁵⁶ tion is twofold: (1) RAFFT for efficient prediction of pseudoknot-
⁵⁷ free RNA folding pathways using the fast Fourier transform; (2)

58 aRNAque, an evolutionary algorithm inspired by Lévy flights for
59 RNA inverse folding with or without pseudoknot.

60 The first tool, RAFFT, implements a novel heuristic to predict
61 RNA secondary structure formation pathways that has two com-
62 ponents: (i) a folding algorithm and (ii) a kinetic ansatz. When
63 considering the best prediction in the ensemble of 50 secondary
64 structures predicted by RAFFT, its performance matches one of the
65 recent deep-learning-based structure prediction methods. RAFFT
66 also acts as a folding kinetic ansatz, which we tested on two
67 RNAs: the coronavirus frameshifting stimulation element (CFSE)
68 and a classic bi-stable sequence. In both test cases, only fewer
69 structures were required to reproduce the full kinetics, whereas
70 known methods required a sample of 20,000 structures.

71 The second tool, aRNAque, implements an Evolutionary Algo-
72 rithm (EA) inspired by the Lévy flights mechanism, which sup-
73 ports pseudoknotted target structures. The number of point mu-
74 tations at every step of aRNAque'EA is drawn from a Zipf distri-
75 bution. Therefore, our proposed method increases the diversity
76 of designed RNA sequences and reduces the average number
77 of evaluations of the evolutionary algorithm. The overall perfor-
78 mance showed improved empirical results compared to existing
79 tools through intensive benchmarks on both pseudoknot and
80 pseudoknot-free datasets.

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

85 PUBLICATIONS

86 This thesis presents our contributions to RNA secondary struc-
87 tures' computational methods for the folding and inverse folding
88 problems. They were obtained in collaboration with my advi-
89 sor Matteo Smerlak, Vaitea Opuu and Vincent Messow. Most of
90 the ideas and figures have appeared previously in the following
91 publications:

- 92 • [112] **Nono SC Merleau** and Matteo Smerlak (2021). *A*
93 *simple evolutionary algorithm guided by local mutations for an*
94 *efficient RNA design.* In: *Proceedings of the Genetic and Evolu-*
95 *tionary Computation Conference.* pp. 1027-1034.
- 96 • [120] Vaitea Opuu, **Nono SC Merleau**, Vincent Messow,
97 and Matteo Smerlak(2021). *RAFFT: Efficient prediction of*
98 *RNA folding pathways using the fast Fourier transform.* In:
99 *bioRxiv* (**Submitted** to PLoS Comput. Biol.)
- 100 • [113] **Nono SC Merleau** and Matteo Smerlak (2022). *An*
101 *evolutionary algorithm for inverse RNA folding inspired by Lévy*
102 *flights.* In: *bioRxiv* (**Submitted and Accepted**) (At BMC Bioin-
103 *formatics*).

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

- 110 • Igor Haman Tchappi, Stéphane Galland, Vivient Corneille
111 Kamla, Jean-Claude Kamgang, **Cyrille Merleau S Nono**,
112 and Hui Zhao (2019). *Holonification model for a multilevel*
113 *agent-based system.* In: *Personal and Ubiquitous Computing*
114 *23(5).*
- 115 • Ivan P Yamshchikov, **Cyrille Merleau Nono Saha**, Igor
116 Samenko, Jürgen Jost (2020). *It Means More if It Sounds*
117 *Good: Yet Another Hypothesis Concerning the Evolution of Poly-*
118 *semous Words.* In: *Proceedings of the 5th International Con-*
119 *ference on Complexity, Future Information Systems and Risk*
120 *(COMPLEXIS 2020), pages 143-148.*

- ¹²¹ • **Nono SC Merleau**, Sophie Pénisson, Philip J Gerrish, San-
¹²² tiago F Elena, and Matteo Smerlak (2021). *Why are viral*
¹²³ *genomes so fragile? The bottleneck hypothesis*. In: *PLoS. Comput Biol.* 17(7).

¹²⁵ will be addressed in this manuscript.

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

— Donald E. Knuth [90]

131 ACKNOWLEDGEMENTS

¹³² Acknowledgements to be put here.

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

[June 17, 2022 at 17:55 – 1.0]

135 CONTENTS

136	1	INTRODUCTION	1
137	1.1	Survey	1
138	1.2	Characteristics and biological functions of non-coding RNAs.	3
140	1.3	Recent advancements in determining ncRNA functions	5
142	1.4	Biochemistry of RNA molecules	6
143	1.5	Bioinformatic definitions.	9
144	1.5.1	Structural definitions	9
145	1.5.2	Thermodynamic definitions	14
146	1.5.3	Structural distance definitions	16
147	1.5.4	RNA folding map properties	18
148	1.6	Conclusion and outline of the thesis	19
149	I	RNA FOLDING	
150	2	INTRODUCTION TO RNA FOLDING	25
151	2.1	Stability and prediction of RNA secondary structures	25
153	2.1.1	MFE prediction tools for pseudoknot-free RNA sequences using a score-base method	27
155	2.1.2	ML-based methods	30
156	2.1.3	Prediction tools for pseudoknotted RNA sequences	32
158	2.2	RNA kinetics	33
159	2.3	Conclusion	37
160	3	RAFFT: EFFICIENT PREDICTION OF FAST-FOLDING PATHWAYS OF RNAs	39
162	3.1	Material and Methods	39
163	3.1.1	RAFFT's algorithm description	39
164	3.1.2	Kinetic ansatz	43
165	3.1.3	Benchmark datasets.	43
166	3.1.4	Structure prediction protocols	44
167	3.2	Experimental results	45
168	3.2.1	RAFFT's run time and scalability	45
169	3.2.2	Accuracy of the predicted structural ensemble	47
171	3.2.3	Applications to the RNA kinetics	49
172	3.3	Conclusion	54

173	II RNA DESIGN	
174	4 INTRODUCTION TO RNA DESIGN	57
175	4.1 RNA inverse folding and biotechnological implications	57
176	4.2 The positive and negative design.	58
177	4.3 Objective functions previously used in the context	
178	of Inverse RNA folding.	59
179	4.4 A review on existing inverse RNA folding tools.	61
180	4.4.1 Pseudoknot-free RNA inverse folding tools	61
181	4.4.2 Pseudoknotted RNA inverse folding tools	66
182	4.5 Benchmarking the Inverse folding tools	67
183	4.6 Conclusion	68
185	5 AN EVOLUTIONARY ALGORITHM FOR INVERSE FOLDING	
186	INSPIRED BY LÉVY FLIGHTS.	71
187	5.1 Material and methods	71
188	5.1.1 aRNAque's mutation operator	71
189	5.1.2 aRNAque's objection functions	74
190	5.1.3 aRNAque's EA	75
191	5.1.4 Benchmark parameters and protocols	76
192	5.2 Experimental results	79
193	5.2.1 aRNAque's performance on pseudoknot-free	
194	target structures	79
195	5.2.2 aRNAque's performance on pseudoknotted	
196	target structures	84
197	5.2.3 Quality of the designed RNA sequences	87
198	5.2.4 Complexity and CPU time comparison	89
199	5.3 Conclusion	91
200	III GENERAL CONCLUSION AND DISCUSSIONS	
201	6 LIMITATIONS OF THE PROPOSED METHODS AND PER-	
202	SPECTIVES	95
203	6.1 RAFFT: Limitations and future works	95
204	6.2 aRNAque: Limitations and perspectives	97
205	6.3 RAFFT and evolutionary dynamics perspectives	101
206	A APPENDIX	105
207	A.1 aRNAque's GC-content parameters	105
208	A.2 Benchmark on Eterna100 dataset	105
209	A.3 General EA benchmark parameters	105
210	A.4 Other benchmark on Eterna100-V1	107
211	A.5 Tools patching	108
212	A.6 aRNAque example calls	109

²¹⁴ LIST OF FIGURES

²¹⁵	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	⁷
²¹⁶			
²¹⁷			
²¹⁸			
²¹⁹	Figure 1.2	RNA base pair interactions. (a) and (b) are commonly know as Watson-Crick base pairs. (c) is the wobble base pair. Hydrogen bonds are indicated as grey dashed lines. Substitution of bold hydrogen residues with ribose 5-phosphate yields the corresponding nucleotides found in RNA molecules.	⁸
²²⁰			
²²¹			
²²²			
²²³			
²²⁴			
²²⁵			
²²⁶	Figure 1.3	RNA secondary structure representation	¹²
²²⁷	Figure 1.4	RNA secondary structure loop decomposition. Each loop is highlighted in blue.	¹³
²²⁸			
²²⁹	Figure 3.1	Algorithm execution for one example sequence which requires two steps. (Step 1) From the correlation $cor(k)$, we select one peak which corresponds to a position lag k . Then, we search for the largest stem and form it. Two fragments, “In” (the interior part of the stem) and “Out” (the exterior part of the stem), are left, but only the “Out” may contain a new stem to add. (Step 2) The procedure is called recursively on the “Out” sequence fragment only. The correlation $cor(k)$ between the “Out” fragment and its mirror is then computed and analyzing the k positional lags allows to form a new stem. Finally, no more stem can be formed on the fragment left (colored in blue), so the procedure stops.	⁴¹
²³⁰			
²³¹			
²³²			
²³³			
²³⁴			
²³⁵			
²³⁶			
²³⁷			
²³⁸			
²³⁹			
²⁴⁰			
²⁴¹			
²⁴²			
²⁴³			
²⁴⁴			
²⁴⁵			

246 Figure 3.2

247

248

249

250

251

252

253

254

255

256

257 Figure 3.3

258

259

260

261

262

263

264 Figure 3.4

265

266

267

268

269

270

271

272

273

274

275 Figure 3.5

276

277

278

279

280

281

282

283 Figure 3.6

284

285

286

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 [30]. [42](#)

Execution time comparisons. For samples of 30 sequences per length, we averaged the execution times of five folding tools. The empirical time complexity $O(L^\eta)$ where η is obtained by non-linear regression. RAFFT denotes the naive algorithm where only $N = 1$ structure can be saved per stack. [46](#)

Impact of the number of positional lags n and the stack size N on the runtime complexity. For a corresponding length, we generated 30 random sequences, and averaged their execution times. Solid lines indicate the estimated time complexity $O(L^\eta)$ where η is obtained with a non-linear regression on these average execution times for (A) Different number of positional lags n and (B) Different stack sizes N . [47](#)

RAFFT's performance on folding task. (A) PPV vs sequence length. In the top panel, RAFFT (in light blue) shows the PPV score distributions when for the structure (out of $N = 50$ predictions) with the lowest free energy, whereas RAFFT* (in blue) shows the best PPV score in that ensemble. (B) Sensitivity vs sequence length. [49](#)

Structure space analysis. PCA for the predicted structures using RAFFT, RNAfold, MxFold2 compared to the known structures denoted "True". [50](#)

287 Figure 3.7

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

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 o). 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. [51](#)

315 Figure 3.8

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

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.

52

334 Figure 3.9

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

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.

53

361 Figure 5.1

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387 Figure 5.2

388

389

390

391

392

393

394

395

396

397

398

399

400

401

Binomial vs. Zipf distributions. (A) Samplings Binomial and Zipf distributions for the best binomial mutation rate μ^* (respectively c^* for the best Zipf exponent parameter). Both distributions have a mean of 8.7 point mutations for a sequence of length 88 nucleotides. (B) Tuning of binomial mutation rate parameter. For each $\mu \in [0, 1]$ with a step size of 0.005 and the pseudoknotted target PKB00342 of length 88, 50 sequences were designed using aRNAque. (B) shows the median generations and the success percentage *vs.* the mutation rate (μ). The best mutation rate is $\mu^* = 0.085$ (with a median number of generation 93.5 and a success rate of 92%). (C) Tuning of Levy exponent. Similar to (B), for each $c \in [0, 7]$ with a step size of 0.1 and for the same pseudoknotted target structure, 100 sequences were designed using aRNAque. It shows the median generations and the percentage of success *vs.* the exponent parameter (c). The Zipf exponent distribution that produced the highest success rate and the minimum number of generations is $c^* = 1.4$. [74](#)

Parameter tuning for both binomial and Lévy mutation schemes. (A) Lévy mutation parameter tuning. Histogram of best exponent parameter (c^*) for a set of 81 target structures with different pseudoknot patterns and various lengths. The most frequent best exponent value is 1. (B) Binomial parameter tuning. Histogram of best mutation rate (μ^*) for the same set of 81 target structures with different pseudoknots and various lengths. The most frequent best parameter is the low mutation rate ($\approx 1/L$). For some structures, the best mutation rate is the high one for different lengths as well. [79](#)

402 Figure 5.3

403

404

405

406

407

408

409

410

411

412

413

414

415

416 Figure 5.4

417

418

419

420

421

Lévy mutation vs. Local mutation: performance analysis with respect to the base-pair density. The higher the base-pair density is, the more useful the Lévy mutation scheme to speed up the optimization EA. (A) Distributions of number of generations for the low and high base-pair density targets of the Eterna100 dataset. (B) Percentages of targets with low and high base-pair density for the Eterna100 and PseudBase++. (C) The length distributions of the low and high base-pair density pseudoknot-free and pseudoknotted targets. [80](#)

aRNAque's performance on a TRIPOD secondary structure. (A) The tripod target structure. (B) aRNAque's solution using the Turner1999 energy parameter sets. (C) aRNAque's solution using the Turner2004 energy parameter sets. [82](#)

422 Figure 5.5

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446 Figure 5.6

447

448

449

450

451

452

453

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.

85

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

86

- 454 **Figure 5.7** **aRNAque vs antaRNA on PseudoBase++ dataset**
 455 **using IPknot: GC-content analysis.** (A)
 456 Base-pair distance ditributions. (B) GC-
 457 content distance distributions. The differ-
 458 ence between the targeted GC-content and
 459 the actual GC-content values. In (A,B),
 460 lower values imply better performance.
 461 (C) Number of successes realised by both
 462 inverse folding tools. Two values are con-
 463 sidered: the up value represent the num-
 464 ber targets successfully solved for each
 465 GC-content value out of the 266 targets
 466 benchmarked; the down values represent
 467 the number sequences folding into the tar-
 468 geted secondary structure. **88**
- 469 **Figure 5.8** **aRNAque vs antaRNA on PseudoBase++ dataset**
 470 **using IPknot: Diversity analysis.** The po-
 471 sitional entropy distributions plotted agains
 472 the targeted GC-content values. Higher
 473 values imply better performance. **89**
- 474 **Figure 5.9** **CPU time: RNAinverse vs. aRNAque.** Each
 475 bubble corresponds to a target structure
 476 in EteRNA100 dataset and, their colours
 477 are proportional to the length of the tar-
 478 gets. In the legend, MHD stands for Me-
 479 dian Hamming distance, and the differ-
 480 ent markers represent—('o') 100% suc-
 481 cess for both tools—('+') 100% success for
 482 aRNAque and not for RNAinverse—('−') for
 483 the case both tools fail to find at least one
 484 sequence that folds into the target. Under-
 485 lying the CPU time difference is the inside
 486 plot that shows the CPU time (in seconds)
 487 as a target length function. **90**

488 Figure 5.10

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503 Figure 6.1

504

505

506

507

508

509

510

511

512

513

514

515 Figure 6.2

516

517

CPU time analysis using Hotknots: antaRNA

vs. aRNAque. Each bubble corresponds to a target structure in PseudoBase++ dataset and, their colours are proportional to the length of the targets. In the legend, BP stands for Median base pair distance, and the different markers represent—('o') 100% success for both tools—('+') 100% success for aRNAque and not for antaRNA—('−') for the case aRNAque's designed sequences are of median base pair distances greater than the one of antaRNA. Underlying the CPU time difference is the inside plot that shows the CPU time (in seconds) with respect to the target length. [92](#)

Lévy mutation vs one-point mutation. For the Eterna100 target structure [*CloudBeta*] *5 Adjacent Stack Multi-Branch Loop*, ten independent runs were performed in which a minimum of 10 sequences were designed per run. (A) Max fitness and mean fitness (inset) over time. (B) Distinct sequences *vs.* Distinct structures over time. (C) Mean Shannon entropy of the population sequences over time for both binomial and Lévy mutation. (D) The max fitness plotted against the entropy over time. [99](#)

Distribution of number of generations need to solve the target T_1 , for both Lévy and Local mutation schemes. [101](#)

518 Figure 6.3

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

Simulation of an RNA population evolving toward a tRNA (See the figure on the right side) target secondary structure. The target was reached after 933 generations (i.e. $\approx 10^5$ replications). The black line shows the average structure distance of the structures in the population to the target structure. The evolutionary history linking the initial structure to the target structure comprises 23. Each structure is labelled by an integer taken from 0 to 22. To each of them corresponds one horizontal line (in red). The top-level corresponds to the initial structure and the bottom the target structure. At each level, a series of red intervals correspond to the periods when the structure was present in the population, and the green curve represents the transition between structures. Only the time axis has a meaning for the red and green curves. [103](#)

539 LIST OF TABLES

540 Table 3.1

541

542

543

544

545

546

547

548

549 Table 5.1

550

551

552

553

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. [48](#)

Summary of performance of aRNAque vs the 7 other algorithms benchmarked on EteRNA100-V1 by Anderson-Lee et al. [3] (using the recent energy parameter sets, the Turner2004) [82](#)

554	Table 5.2	Summary of performance of aRNAque vs the 10 other algorithms benchmarked on the non-EteRNA100 by Anderson-Lee et al. [3] 83
555		
556		
557		
558	Table A.1	Mutation parameters used in aRNAque to control the GC-content values. 105
559		
560	Table A.2	Evolutionary algorithm parameter for each benchmarks. 106
561		
562	Table A.3	Different parameters for the base pair distributions 107
563		
564	Table A.4	Success percentage on Eterna100 datasets for each set of mutation parameters. 108
565		

566 LISTINGS

567 ACRONYMS

568

[June 17, 2022 at 17:55 – 1.0]

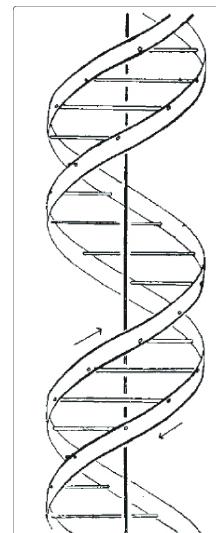
569

570 INTRODUCTION

571 1.1 SURVEY

572 DNAs and RNAs are macromolecules in the nucleus of eukaryotic
573 cells that allow storing information with the help of nucleotides.
574 Nucleotides consist of a five-carbon sugar, a phosphate group,
575 and a nucleobase. There are four nucleotides in the DNA, distin-
576 guished by their nucleobase: A for Adenine, T for Thymine, G for
577 Guanine, and C for Cytosine. Similar to DNA, we also find four
578 different nucleotides in RNA, distinguished by the nucleobase
579 with only one exception; the Uracil (U), which replaces Thymine
580 in DNA. Even though the basis blocks constituting the DNA were
581 known for many years, in 1953, James Watson and Francis Crick
582 [184] succeeded in putting them together and suggested a rea-
583 sonable DNA structure. Their work revealed for the first time that
584 the structure of DNA molecules has helical chains, each coiled
585 round the same axis where the chain consists of phosphate di-
586 eter groups. The two chains are held together by the purines and
587 pyrimidine bases; they are joined together in pairs, a single base
588 from the other chain bonded to a single base from the other chain.
589 For the bonding to occur, one of the pairs must be Adenine and
590 thymine or Guanine and Cytosine. A DNA molecule structure is
591 depicted on the left side of the page. In contrast to DNA, RNAs are
592 mostly single-stranded, and the complementary pairings formed
593 in the structure are A-U, G-U and G-C.

594 Watson and Crick's elucidation of DNA structure has moti-
595 vated many other scientists to investigate further the structural
596 implications of molecules in functions such as replication and
597 gave rise to modern molecular biology. Later in the same year,
598 Crick formulated the central dogma of molecular biology that
599 describes the flow of information between DNA, RNA, and pro-
600 teins [25]. He described the dogma in two steps: from DNA
601 to mRNA through transcription and from mRNAs to proteins
602 through translation. Since this central dogma was proposed, more
603 works have been done to investigate each step. DNAs are tran-
604 scribed into RNA molecules (messenger RNAs) that contain the
605 same information as the template DNAs. Subsequently, these
606 RNA messengers are translated into proteins according to the



Helical representation of DNA structures [184].



The tertiary structure of tRNA. The CCA-tail is in yellow, the acceptor stem in purple, the variable loop in orange, D-arm in red, the anticodon arm in blue with anticodon in black, and T-arm in green (Taken from Wikipedia) [47, 140].

The function of non-coding RNAs is largely determined by their high-dimensional structure [19]. For instance, we can analyze the catalytic function of ribozymes in terms of basic structural motifs, e.g. hammerhead or hairpin structures [37]. Other RNAs, like riboswitches, involve changes between alternative structures [178]. Understanding the relation sequence and structure is a central challenge in molecular biology. In the last 20 years, many different methods for determining the RNA structures of molecules have emerged: from experimental lab methods to computational approaches. For experimental lab methods, X-ray crystallography and the nuclear magnetic resonance (NMR) are the most accurate approaches to offer structural information at a single base-pair resolution. Both experimental methods are often characterized by high experimental cost and low throughput. In addition to those limitations, RNA molecules are volatile and difficult to crystallize. Despite the development of more sophisticated techniques to infer the state of nucleotides in RNA molecules using enzymatic [82, 176] or chemical probes [169, 187] coupled with next-generation sequencing [11, 168], most of them can only capture RNA structures *in vitro* which mostly differ from the *in vivo* structure conformations. Experimentally, only a tiny fraction of known ncRNAs has been determined [124]. Because measuring the structure of RNAs experimentally is very difficult and expensive, computational approaches play a central role in the analysis of natural RNAs [46, 147], and are an essential alternative to

650 experimental approaches. RNAs fold into secondary structures
651 before folding into higher-level (tertiary and quaternary) struc-
652 tures [16, 170]. This separation of time scales justifies focusing
653 on the secondary structure prediction; evidence suggests that
654 the RNA's secondary structures largely determine the resulting
655 high-level structures.

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

666 1.2 CHARACTERISTICS AND BIOLOGICAL FUNCTIONS OF NON-
667 CODING RNAs.

668 In the previous section, we introduced the central dogma of mi-
669 crobiology, which describes the flow of information in the living
670 systems. Two important non-coding RNAs involved in the pro-
671 tein machinery have been highlighted. In this section, we provide
672 some of the main characteristics of ncRNAs, and we emphasize
673 how those characteristics often play an essential role in realizing
674 their functions.

675 What motivates the computational studies of ncRNAs is often
676 the importance of the biological function they play. Consequently,
677 the ncRNAs can be classified based on their biological functions.
678 Although many recent transcriptomic and bioinformatic studies
679 suggested thousands of ncRNAs with their functional importance,
680 the total number of ncRNAs encoded in the human genome still
681 remains unknown [140]. More recently, newly identified ncRNAs
682 have not been validated by their function; it could be possible
683 that most of them are non-functional. Some experiments *in vitro*
684 evolution have shown that RNA molecules can catalyze various
685 chemical reactions relevant to biological processes such as RNA
686 replication, nucleotide synthesis, thymidylate synthesis, lipid
687 synthesis, and sugar metabolism [42, 134]. Another characteris-
688 tic of ncRNAs is their lengths formed post-transcriptionally. We
689 often distinguish two main ncRNA classes of critical biological
690 functions: the short non-coding RNAs (sncRNAs with length

691 < 30nt) and the long non-coding RNAs (lncRNAs with length
692 > 200nt). The length limit is often because of the practical consid-
693 erations, including separating RNAs in standard experimental
694 protocols. The length of non-coding RNAs is also taken into ac-
695 count in computational studies, and it will be used throughout
696 our work to distinguish RNA sequences and structures in the
697 different datasets considered.

698 The function of lncRNAs includes a role in higher-order chro-
699 mosomal dynamics, telomere biology, and subcellular structural
700 organization [10, 26]. Some lncRNAs play key regulatory and
701 functional roles in the gene expression program of the cell. One of
702 the vital functions is to act as ribozymes. Examples of naturally oc-
703 curring ribozymes include group I and group II introns—RNase
704 P and the hammerhead. The group, I and group II introns are
705 usually 200 – 600nt long, catalyzing RNA splicing. Many sncR-
706 NAs also contribute to the realization of similar biological func-
707 tions. For example, small interfering RNAs contribute to gene
708 regulation, transposon control and vital defence. MicroRNAs par-
709 ticipate in the post-transcriptional gene regulation, microRNA-
710 offset RNAs (moRNAs), PIWI-interacting RNAs (piRNAs) and
711 promoter-associated RNAs (PARs) contribute to the gene regula-
712 tion. More recently, many discoveries revealed several non-coding
713 RNAs implicated in cancer growth and MCL-1 expression regu-
714 lation [140, 181]. Those examples include ncRNAs from different
715 classes, miRNAs, snoRNA and T-UCR, all associated with a spe-
716 cific disease [45, 140].

717 There are also other classes of ncRNAs such as Aptamers and
718 riboswitches that have also been observed in nature. Aptamers
719 are ncRNAs that can bind to other specified targets, whose na-
720 ture is highly diverse. They range from small molecules to larger
721 molecules. In some contexts, aptamers are termed riboswitches;
722 for example, when their function is to sense the presence of an
723 associated metabolite to cause a specific cis-reaction and/or cis-
724 regulation of subordinated functional pathways [188].

725 In sum, lnc/snc-RNAs contribute to the realization of various
726 biological functions, and they are mostly distinguishable based on
727 their length. Their characteristics in terms of lengths are primarily
728 due to the practical considerations in the standard experiments.
729 But, their functions allow us to distinguish them better. In the
730 next section, we provide some of the recent advancements in the
731 techniques used to identify functional ncRNAs.

732 1.3 RECENT ADVANCEMENTS IN DETERMINING NCRNA FUNC-
733 TIONS

734 Most of the previously mentioned functions of ncRNAs are iden-
735 tified using gene targeting techniques, a well-known technique
736 used to investigate protein functions [143]. In addition, exper-
737 imental approaches are used to define ncRNA functions. With
738 the recent advancements in genome engineering, a method such
739 as Clustered Regularly Interspaced Short Palindromic Repeats
740 (CRISPR) has been employed to tag lncRNAs, allowing to capture
741 specific RNA-protein complexes assembled *in vivo*.

742 The CRISPR [8] was described by Barrangou and his collab-
743 orators in 2007 as a distinctive genome feature of most bacteria
744 and archaea and thought to be involved in resistance to bacte-
745 riophages. It is an adaptive defence system against viruses and
746 plasmid intrusions. When a successful defence takes place, the
747 system updates information about the intruder's genetic material.
748 This update will then allow the system's host to identify its en-
749emy, making it robust and durable in the future. The information
750 about the intruder's genetic material is stored in short repeating
751 stretches of RNA, which can, in the case of a new intrusion, be
752 incorporated into a carrier protein(CAS). The capacities of the
753 CRISPR/CAS9 of selectively destroying foreign DNA/RNA and
754 editing the genome was identified by Li et al. [93], and it was
755 turned into methods allowing to alter and edit single genes within
756 genomes selectively. The same technology is also successfully ap-
757 plied to animal cell lines [74, 80, 180] and industrial plants [94,
758 162].

759 Another method of SELEX (Systematic Evolution of Ligands
760 by Exponential Enrichment) [172] introduced by Tuerk in the
761 early 1990s offers the possibility of enriching stretches of RNA
762 that can bind towards a certain target. The method relies on
763 mechanisms usually ascribed to the process of evolution, that
764 is, variation, selection, and replication. A pool of RNAs that are
765 entirely randomized at specific positions is subjected to selection
766 for binding, in this case to GP43 on nitrocellulose filters. The
767 selected RNAs are amplified as double-stranded DNA competent
768 for subsequent in vitro transcription. This newly transcribed RNA
769 is enriched for better binding sequences and is then subjected to
770 selection to begin the next cycle. Multiple rounds of enrichment
771 result in the exponential increase of the best binding ligands
772 until they dominate the population of sequences. SELEX has
773 given rise to numerous synthetic aptamers with different targets

774 in its application. They have been subject to a further extension
 775 towards inclusion into regulative RNA entities.

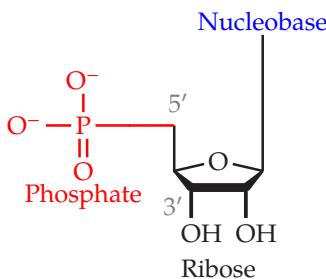
776 More recently, increased types of ncRNAs have been detected
 777 and identified by the development of next-generation sequencing
 778 (NGS) [182], which can be roughly divided into the process sec-
 779 tions of sample preprocessing, library preparation, sequencing,
 780 and bioinformatics.

781 The functions of many ncRNAs are dependent on their high-
 782 level structures, which often depend on a low-level ones such as
 783 secondary structures. Knowing the RNA structure of an ncRNA
 784 plays a vital role in probing its function. For example, it can help
 785 interpret experiments relating to the mechanism of RNA func-
 786 tion [52]. Or, it can help propose new experiments to probe func-
 787 tion [83]. Therefore, understanding even the secondary structure
 788 alone can assist both of these examples. In the following section,
 789 we provide a biochemical definition of the secondary structure
 790 of ncRNAs and an overview of the different interactions involved
 791 during their formation.

792 1.4 BIOCHEMISTRY OF RNA MOLECULES

793 So far in this work, we provided a biological motivation for study-
 794 ing non-coding RNA as an independent entity. The discovery
 795 of new ncRNAs functions has emerged through intensive exper-
 796 imental studies and with recent advanced techniques in next-
 797 generation sequencing. Several examples demonstrated the im-
 798 portance of the ncRNA structures in the probing process of new
 799 functions. The process in which RNA sequences are mapped to
 800 their corresponding structures is called RNA folding. In nature,
 801 this process is thought to be hierarchical [16, 170]. Nucleotides
 802 form a chain given their sequence of bases (primary structure);
 803 RNAs fold into secondary structures, such as stem-loops and
 804 helices, before folding into higher-level (tertiary and quaternary)
 805 structures. Our work is restricted here to the secondary level of
 806 an RNA structure, i.e., the set of canonical pairs. This section
 807 provides a biochemical definition of different nucleotides and
 808 base-pair interactions involved in the secondary structure folding
 809 of RNA molecules.

810 Chemically, each nucleotide in RNA molecules consists of a
 811 phosphate residue, a pentose sugar and a nucleobase. The typical
 812 chemical structure of a nucleotide is depicted on the right side of
 813 the page. Figure 1.1 illustrates the chemical structure of each of
 814 the four different nucleobases found in RNA (A, C, G and U). A



Structure of an RNA nucleotide

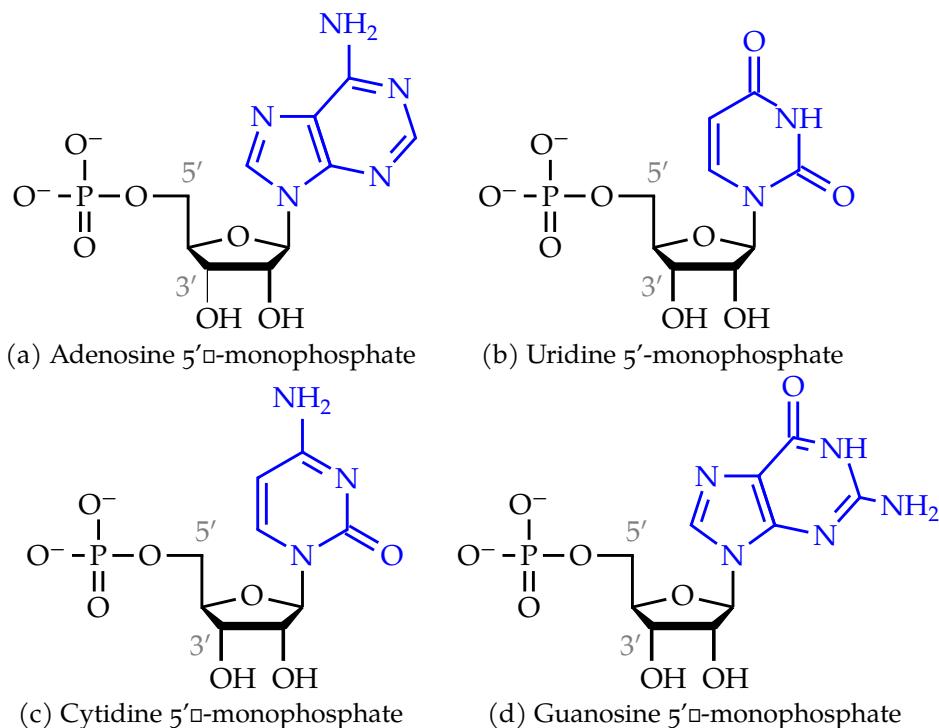
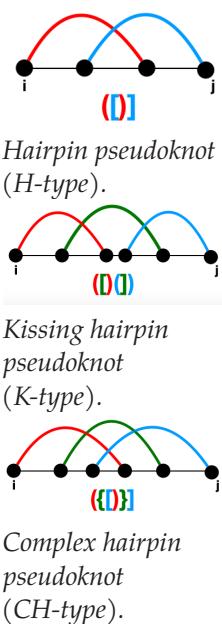


Figure 1.1: RNA nucleotides. Adenine and guanine belongs into the chemical class of purine molecules and the uracil and thymine in the class of pyrimidines

815 nucleotide is a nucleoside which has a (mono, di, trip) phosphate
 816 residue bound to its 5'-carbon atom. By convention, the carbon
 817 atoms of the pentose sugar in nucleotides are numbered with
 818 *primes*.

819 At the lowest level, RNA molecules are simply represented as
 820 a list of nucleobase characters. The 5'-3' phosphodiester bonds
 821 attach the different nucleotides composing the RNA molecule be-
 822 tween ribose to form the primary structure of RNA. The chain di-
 823 rection is conventionally designed as 5' to 3' (i.e. from 5'-phosphate
 824 first sugar backbone to the 3'-hydroxyl last sugar in the sequence).

825 In contrast to the RNA primary structure, the secondary struc-
 826 ture consists of a list of nucleobase pairs, and the hydrogen bonds
 827 between the bases form base pairs. Different interactions are pos-
 828 sible between the bases depending on the structure level con-
 829 sidered. At the secondary level, we have the Watson-Crick (or
 830 canonical) pairs [135, 146] (A-U and G-C), the Wobble (or non-
 831 canonical) (G-U) pairs that occur with reduced frequency. Figure
 832 1.2 shows the chemical base pairs for the Watson-Crick and Wob-
 833 ble interactions.



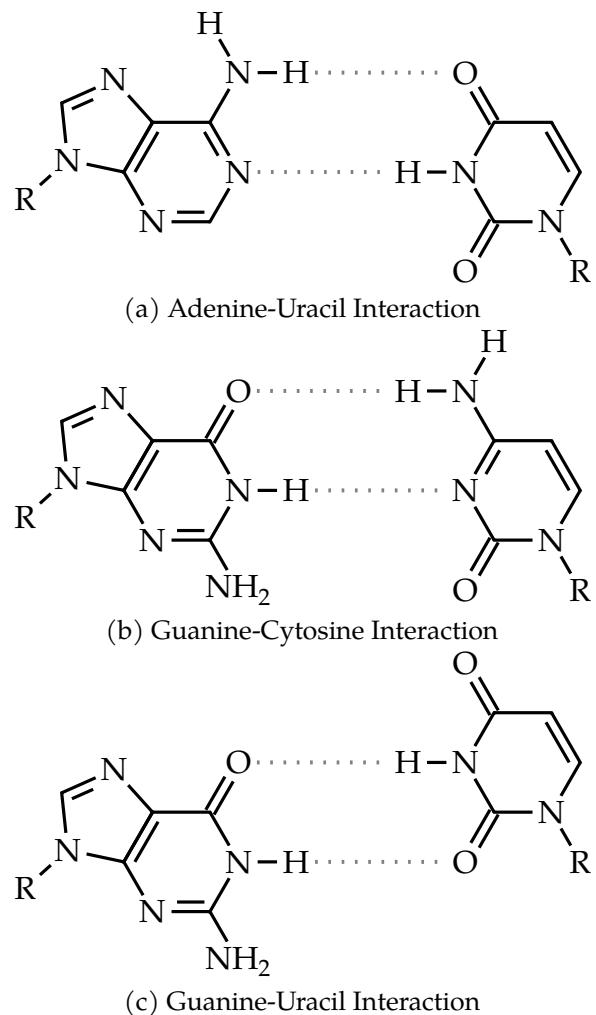


Figure 1.2: **RNA base pair interactions.** (a) and (b) are commonly known as Watson-Crick base pairs. (c) is the wobble base pair. Hydrogen bonds are indicated as grey dashed lines. Substitution of bold hydrogen residues with ribose 5-phosphate yields the corresponding nucleotides found in RNA molecules.

We also find crossing or pseudoknotted interactions in natural RNA that play vital roles in realising biological functions. Pseudoknots occur when two canonical or non-canonical interactions cross each other [186]. The three type of pseudoknot patterns often in RNA are depicted on the left side. Even though pseudoknots are often considered the beginning of the interaction between the secondary and tertiary levels of RNA structures, we consider them 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 permitted. The following section will provide formal definitions and the framework in which the folding of the secondary structure of ncRNAs can be computationally studied.

1.5 BIOINFORMATIC DEFINITIONS.

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

1.5.1 Structural definitions

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

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

$$\phi = (\phi_1, \dots, \phi_L), \quad (1.1)$$

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

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

- 874 1. A nucleotide (sequence position) can only belong to a single
 875 pair, i.e. $\forall (i, j), (k, l) \in \mathcal{S}$ with $i < k : i = k \Rightarrow j = l$.
- 876 2. Paired bases must be separated by at least three unpaired
 877 nucleotides. i.e. $\forall (i, j) \in \mathcal{S} \Rightarrow j - i \geq 3$.
- 878 3. There are no pseudoknots, i.e. $\nexists (i, j), (k, l) \in \mathcal{S}$ with $i <$
 879 $k < j < l$,
- 880 4. The base pairs consist exclusively of Watson–Crick (C–G
 881 and A–U) pairs and Wobble (G–U) pairs. i.e. $\forall (i, j) \in \mathcal{S} \Rightarrow$
 882 $\phi_i \phi_j \in \{GC, CG, AU, UA, GU, UG\}$,

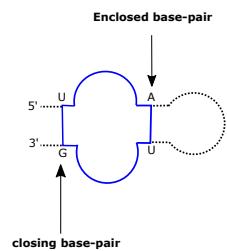
883 **DEFINITION 3** (Secondary structure representation): A graph-
 884 ical way of representing an RNA secondary structure. There are
 885 several representations of \mathcal{S} .

- 886 • Dot-bracket (or string) representation: In this representa-
 887 tion, the secondary structure \mathcal{S} is compactly stored in a
 888 string σ consisting of dots and matching brackets. i.e. σ is
 889 a string of length L over the alphabet $\Delta_\sigma = \{(.,), [.,], \{., \}, <$
 890 $, >, .\}$ where, at each unpaired positions we have a dot ‘.’ at
 891 the corresponding string position, and $\forall (i, j) \in \mathcal{S}$, we have
 892 an opening bracket at position σ_i and a closing bracket at
 893 position σ_j . We denote σ the string representation of the
 894 structure \mathcal{S} . Figure 1.3D shows an example of a string rep-
 895 resentation.
- 896 • Planar representation: it is the common way of representing
 897 an RNA secondary structure in which \mathcal{S} is presented as a
 898 graph with each vertex representing a nucleotide and an
 899 edge connecting consecutive nucleotides and base pairs
 900 (See Figure 1.3B).
- 901 • Circular (or circle) representation: similar to planar repre-
 902 sentation, \mathcal{S} is a graph but drawn in the plane in such a way

903 that all vertices are arranged on a circle, and the edges repre-
 904 senting base pairs lie inside the circle. In a pseudoknot-free
 905 secondary structure circular representation, the edges do
 906 not intersect (See Figure 1.3A).

- 907 • Linear representation: In this representation, \mathcal{S} is a graph in
 908 which the nucleotides are arranged consecutively in a line
 909 and the edges representing base pairs form semi-circle that
 910 do not intersect for pseudoknot-free structure (See Figure
 911 1.3C).
- 912 • Mountain representation: it is mainly used for representing
 913 large structures. \mathcal{S} is presented in a two-dimensional graph,
 914 in which the x -coordinate is the position i of the nucleotide
 915 in the sequence ϕ and the y -coordinate the number $m(i)$ of
 916 base pairs that enclose nucleotide i .
- 917 • Tree representation: \mathcal{S} is drawn as a tree in which internal
 918 nodes are the base pairing positions, and the leaves are the
 919 unpaired positions. The dot-bracket representation is also
 920 often considered as a tree represented by a string of paren-
 921 thesis (base pairs) and dots for the leaf nodes (unpaired
 922 nucleotides).
- 923 • Shapiro representation: it allows representing the different
 924 elements composing \mathcal{S} by single matching brackets, and
 925 the components are labelled with H(Hairpin), B(Bulge), I
 926 (interior loop), M (multi-loop) and S (stacking loop) [149].

927 Figure 1.3 shows some examples of RNA secondary structure rep-
 928 resentation. For graphical illustrating examples in the thesis, we
 929 will mostly use the planar representation, and for computational
 930 methods will use the dot-bracket representation for simplicity.



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

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

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

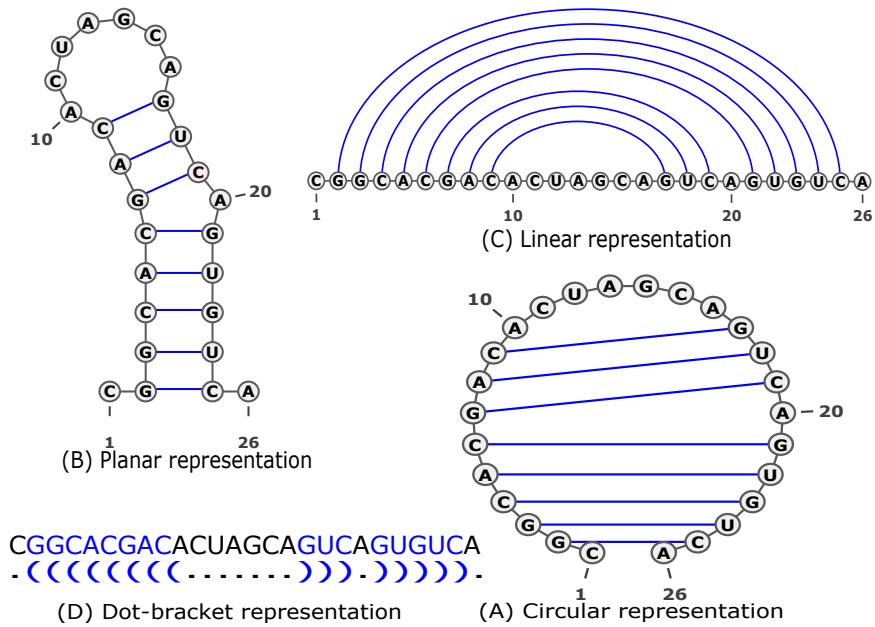


Figure 1.3: RNA secondary structure representation

- 942 1. Interior loop: a loop with degree $d = 2$ i.e $|\mathcal{L}_p| = 2$ and
943 $\mathcal{L}_u \subset \{1, 2, \dots, L\} \cup \emptyset$.
- 944 2. Stacking pair: an interior loop of length $l = 0$ i.e. $|\mathcal{L}_p| = 2$
945 and $\mathcal{L}_u = \emptyset$.
- 946 3. Hairpin Loop: Any loop of degree $d = 1$ and length $l \geq 3$.
947 i.e $|\mathcal{L}_p| = 1$ and $\mathcal{L}_u \neq \emptyset$.
- 948 4. Bulge loop: a special case of interior loop in which there are
949 unpaired bases only on one side. i.e $\mathcal{L}_p = \{(i_1, j_1), (i_2, j_2)\}$
950 with $i_1 \neq i_2, j_1 \neq j_2$ one of the following assumption holds:
 - 951 • If $\exists i' \in \mathcal{L}_u: i_1 < i' < j_2 \Rightarrow \nexists k' \in \mathcal{L}_u: i_2 < k' < j_2$
 - 952 • If $\exists k' \in \mathcal{L}_u: i_2 < k' < j_2 \Rightarrow \nexists i' \in \mathcal{L}_u: i_1 < i' < j_1$
- 953 5. Multi-loop: Any loop with degree $d > 2$ i.e. $|\mathcal{L}_p| \geq 3$ and
954 $\mathcal{L}_u \neq \emptyset$.
- 955 6. Exterior loop: a loop in which all the positions are not inte-
956 rior of any pair i.e. $\mathcal{L}_p = \emptyset$ and $\mathcal{L}_u \neq \emptyset$.

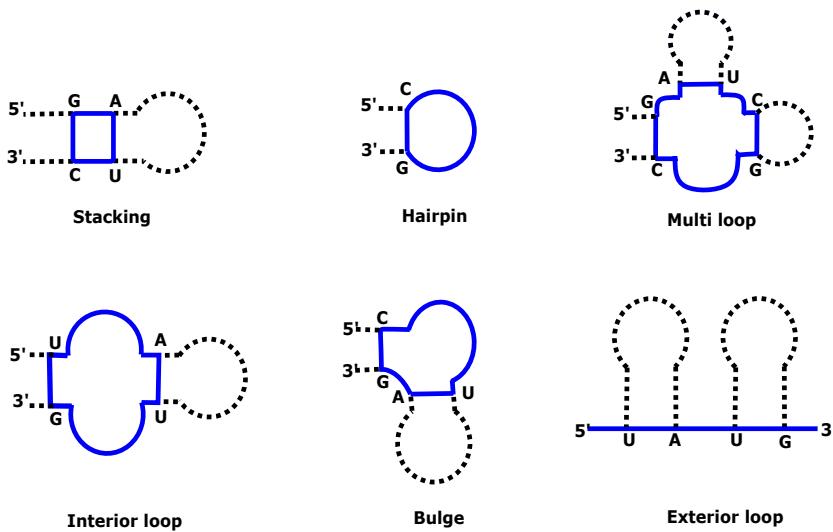


Figure 1.4: **RNA secondary structure loop decomposition.** Each loop is highlighted in blue.

DEFINITION 5 (Free Energy of an RNA secondary structure): Given the loop set $\mathbb{L}_{\phi, \mathcal{S}}$, the free energy ΔG of \mathcal{S} defines its thermodynamic stability. ΔG is the free energy difference with respect to the completely unfolded state [171]. $\Delta G(\mathcal{S}, \phi)$ is computed using the additivity principle [33], by summing up the energies of its constituent loops.

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

Many models allow for computing the free energies of those constituent loops, but the dominant is the nearest-neighbor loop energy model [174]. This model associates tabulated free energy values to loop types and nucleotide compositions; the Turner2004 [106] is one of the most widely used parameter sets.

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

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

where ΔH is the (pressure- and volume-dependent) enthalpy change, T the absolute temperature and ΔS the entropy change. The dominant stabilizing effect is attributed to consecutive base pairs (The stacking loops), whereas long unpaired regions enclosed between base pairs have destabilizing effects [54, 70]. As

a simplified example, the destabilizing free energy contribution $\Delta G(\mathcal{L}_m)$ of a multiloop \mathcal{L}_m as seen in 1.4C is modelled as:

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

where b is the number of all surrounding base pairs and u the number of base pairs [34]. The structure decomposition and the tabulated energy parameter sets allow an efficient dynamic programming algorithm to determine a sequence's minimum free energy (MFE) structure in the entire structure space. A literature review of tools using such techniques is given in Chapter 2 of the thesis.

1.5.2 Thermodynamic definitions

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

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

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

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

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

DEFINITION 7 (Partition function of RNA): Given the free energy change $\Delta G(\mathcal{S})$ of a structure \mathcal{S} , the partition function $Z(\Sigma_\phi)$

995 is defined on the Boltzmann ensemble (or structure ensemble)
 996 of all possible structures of a given sequence ϕ and we write:

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

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

999 **DEFINITION 8** (Secondary structure probability): How prob-
 1000 able is an RNA secondary structure $\mathcal{S} \in \Sigma_\phi$ for the sequence ϕ ?
 1001 Given the free energy change $\Delta G(\mathcal{S})$ of a structure \mathcal{S} , the boltz-
 1002 mann distribution describes the structure's probability at constant
 1003 temperature T among all other possible structure of the same se-
 1004 quence ϕ . The probability $p(\mathcal{S}|\phi)$ depends on the free energy
 1005 $\Delta G(\mathcal{S})$, the lower the more probable. We write:

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

1006 where, Z is the partition function and $\beta = (RT)^{-1}$ the thermal
 1007 constant.

1008 **DEFINITION 9** (MFE secondary structure): To predict bio-
 1009 logically relevant structures, most computational methods search
 1010 for structures that minimize the free energy. For a given sequence
 1011 ϕ , let Σ_ϕ be the secondary structure ensemble of ϕ . The mini-
 1012 mum free energy structure \mathcal{S}_{MFE} is the structure with the lowest
 1013 probability $p(\mathcal{S}|\phi)$ i.e. the most stable conformation in the ther-
 1014 modynamic equilibrium. We write:

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

1015 **DEFINITION 10** (Base pair probability): Let $\phi = (\phi_i)_{1 \leq i \leq L}$
 1016 be an RNA sequence. The base pair probability matrix $\mathbf{P}(\phi)$ quan-
 1017 tifies the equilibrium structural features of the ensemble Σ_ϕ , with
 1018 entries $P_{i,j}(\phi) \in [0, 1]$ defines as follows:

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

1019 $P_{i,j}(\phi)$ corresponds to the probability that base pair i,j forms at
 1020 the equilibrium. $\mathbf{S}(\mathcal{S})$ is the structure matrix with entries $S_{i,j} \in$
 1021 $\{0, 1\}$. If the structure \mathcal{S} contains pair (i, j) , then $S_{i,j}(\mathcal{S}) = 1$ other-
 1022 wise $S_{i,j}(\mathcal{S}) = 0$.

1023 1.5.3 *Structural distance definitions*

1024 The validation of the results obtained in this thesis is purely
 1025 empirical. We achieved this goal by comparing the predicted and
 1026 expected structures for the folding tools. We use the PPV and
 1027 the sensitivity's statistical properties for the benchmark results
 1028 presented in Chapter 3. For the inverse folding tools, we compare
 1029 the MFE structure of the designed sequence to the target structure.
 1030 For that end, a rigorous definition of a measure of similarities
 1031 between two structures is needed. This subsection defines the
 1032 different similarity measurements used throughout this work. In
 1033 addition, it defines the objective functions used in our inverse
 1034 folding presented in Chapter 5 (Definitions 16 and 17).

1035 **DEFINITION 11** The positive predictive value (PPV): it mea-
 1036 sures the fraction of correct base pairs in the predicted structure
 1037 and it is defined as follows:

$$PPV = \frac{TP}{TP + FP} \quad (1.9)$$

1038 where TP and FP stand respectively for the number of correctly
 1039 predicted base pairs (true positives), and the number of wrongly
 1040 predicted base pairs (false positives).

DEFINITION 14 (Sensitivity): it measures the fraction of base
 pairs in the accepted structure that are predicted.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (1.10)$$

1041 where FN stands for the number of base pairs not detected (false
 1042 negatives).

DEFINITION 12 (Base pair distance): Let σ_1 and σ_2 be two
 secondary structures in their string representation. The base pair
 distance between σ_1 and σ_2 is defined as follows:

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

where,

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

1043 **DEFINITION 13** (Hamming Distance): Let σ_1 and σ_2 be two
1044 secondary structures in their string representation. We define
1045 the hamming distance between σ_1 and σ_2 , $d_h(\sigma_1, \sigma_2)$, to be the
1046 number of position where σ_1 and σ_2 differ.

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

where,

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

1047 **DEFINITION 14** (Ensemble defect (ED)) [194]: Given an
1048 RNA sequence ϕ of length L , the ensemble defect \mathcal{D}_E is the ex-
1049 pected base pair distance between a target structure \mathcal{S}^* and a
1050 random structure generated with respect to the Boltzmann prob-
1051 ability distribution. It is defined as follows:

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

1052 where $P_{i,j}$ is the base pair probability matrix entrances, $d_{bp}((\mathcal{S}, \mathcal{S}^*))$
1053 is the base pair distance between two structures, and $\mathbf{S}(\mathcal{S})$ is the
1054 structure matrix with entries $S_{i,j} \in \{0, 1\}$. If the structure \mathcal{S} con-
1055 tains pair (i, j) , then $S_{i,j}(\mathcal{S}) = 1$ otherwise $S_{i,j}(\mathcal{S}) = 0$.

1056 **DEFINITION 15** Normalized Energy Distance (NED): the
1057 difference between the energy of a given sequence ϕ evaluated to
1058 fold into a target structure \mathcal{S}^* and the minimum free energy of the
1059 sequence in its structural ensemble Σ_ϕ . The value is normalized
1060 over all the sequences in a given population P .

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

where,

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

and,

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

1061 1.5.4 RNA folding map properties

1062 This work considers RNA molecule folding and inverse folding
 1063 optimisation problems. In both cases, It is fundamental to define
 1064 the fitness landscape notion. This subsection provides the formal
 1065 definitions of the fitness landscape and examples related to the
 1066 folding and inverse problem. Some properties such as neutrality,
 1067 mutation mode or move operator are also provided. The size of
 1068 the RNA structural ensemble has been analytically computed
 1069 through tools developed by Stein and Waterman [161], and it
 1070 yields an upper bound of $S_L \approx 1.48 \times L^{-\frac{3}{2}} 1.85^L$ structure vis-a-
 1071 vis 4^L sequences. Compared to the total number of sequences,
 1072 the number of structures is much smaller, which means there is
 1073 a high possibility that many sequences fold into the same MFE
 1074 secondary structure. In case that happens, we call the set of those
 1075 sequences a neutral set. The fraction of such sequences defines
 1076 the neutrality of a fitness landscape.

1077 **DEFINITION 1.6** (Fitness landscape) : A fitness landscape \mathfrak{L}
 1078 results from the combination of three elements: a set of configu-
 1079 rations \mathcal{V} , a cost or fitness function f , and a *move* operator ψ that
 1080 induces a topology on the set of configurations. We write:

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

1081 where \mathcal{G}_f is the the landscape underlying the hypergraph whose
 1082 vertices are the elements from \mathcal{V} labelled with values given by f ,
 1083 and whose edges are specified by the move operator ψ .

1084 The fitness function f assigns to each configuration $v \in \mathcal{V}$ a real
 1085 value taken from an interval $\mathbb{I} \subset \mathbb{R}$ as follows:

$$f : \mathcal{V} \rightarrow \mathbb{I}$$

1086 An example of fitness function in the case of inverse folding
 1087 is defined in Chapter 5 (Section 5.1.2), which uses the hamming

1088 distance d_h and $\mathcal{V} = \{A, C, G, U\}^L$. But in this case, the fitness
 1089 defined in the structural space Σ_ϕ . i.e. we have an intermediate
 1090 folding function $\Delta G(\mathcal{S}, \phi)$, mapping any sequence $\phi \in \mathcal{V}$ to an
 1091 MFE secondary structure.

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

$$\psi: \mathcal{V} \rightarrow \mathcal{V}$$

1092 **DEFINITION 17** (Mutation mode): Let $\phi, \phi' \in \mathcal{V} = \{A, C, G, U\}^L$,
 1093 be two RNA sequences. ϕ' is said to be an n -point mutation of ϕ
 1094 if it differs from ϕ at n nucleotides; i.e. $d_h(\phi, \phi') = n$ where $d_h(., .)$
 1095 is the hamming distance on $\{A, C, G, U\}^L$.

1096 A mutation mode is a random variable U taking values in
 1097 $\{1, \dots, L\}$. $P(U = n)$ is defined as the probability that, exactly n
 1098 nucleotides, selected uniformly at random undergo point mutation
 1099 during a mutation event. U can generally be any probability
 1100 distribution.

1101 **DEFINITION 18** (Neutral set of RNA sequences): For a give
 1102 fitness landscape $\mathcal{L} = (\mathcal{G}_f, f, \psi)$, with $\mathcal{V} = \{A, C, G, U\}^L$, two RNA
 1103 sequence ϕ_1 and ϕ_2 are set to be neutral $\iff f(\phi_1) = f(\phi_2)$. We
 1104 call a set $\Gamma \subset \mathcal{V}$ of all such RNA sequences a neutral set. In the
 1105 case of inverse folding, ϕ_1 and ϕ_2 are neutral if they share the
 1106 same MFE secondary structure. In contrast, ϕ_1 and ϕ_2 have the
 1107 same free energy in the folding problem context.

1108 **DEFINITION 19** (Neutral Network): Let $\mathcal{G}(\mathcal{V}, E)$ be a con-
 1109 nected graph in which vertices are all in the neutral sequence set
 1110 Γ (i.e. $\mathcal{V} \subset \Gamma$). \mathcal{G} is said to be a neutral network $\iff \forall e(v_i, v_j) \in E$,
 1111 v_i, v_j differ by a single nucleotide (i.e. $d_h(v_i, v_j) = 1$).

1112 1.6 CONCLUSION AND OUTLINE OF THE THESIS

1113 This introductory chapter presents nucleic acids in general and,
 1114 in particular, a description of RNA and its chemical, biological,
 1115 and algorithmic definitions. Those concepts with biological moti-
 1116 vations constitute the basis of the thesis.

1117 We organize the next part of the thesis into five chapters. The
 1118 two first chapters are grouped into a first result part which only
 1119 concerns the RNA folding. The second part discusses inverse
 1120 folding, and similarly to the first part, it contains two chapters.
 1121 The last chapter discusses the presented results and concludes

1122 by providing some limitations and possible future research di-
1123 rections.

1124 In this first result part, Chapter 2 provides a brief literature
1125 review on the existing computational methods for RNA folding.
1126 The review focuses on thermodynamic and machine learning
1127 methods such as RNAfold, LinearFold and Mfold. The methods
1128 presented in Chapter 2 have some limitations, such as the compu-
1129 tational time, and in some cases, the predicted thermodynamic
1130 structure does not match the native one. Chapter 3 presents our
1131 proposed folding tool called RAFFT, which aims at overcoming
1132 those limitations. RAFFT implements a novel heuristic to predict
1133 RNA secondary structure formation pathways that has two com-
1134 ponents: (i) a folding algorithm and (ii) a kinetic ansatz. This
1135 heuristic is inspired by the kinetic partitioning mechanism, by
1136 which molecules follow alternative folding pathways to their
1137 native structure, some much faster than others. RAFFT starts by
1138 generating an ensemble of concurrent folding pathways ending in
1139 multiple metastable structures, which contrasts with traditional
1140 thermodynamic approaches that find single structures with min-
1141 imal free energies. When analyzing 50 predicted folds per se-
1142 quence, we found near-native predictions for RNAs of length
1143 ≤ 200 nucleotides, matching the performance of current deep-
1144 learning-based structure prediction methods. RAFFT also acts as
1145 a folding kinetic ansatz, which we tested on two RNAs: the coro-
1146 navirus frameshifting stimulation element (CFSE) and a classic
1147 bi-stable sequence. For the CFSE, an ensemble of 68 distinct struc-
1148 tures computed by RAFFT allowed us to produce complete folding
1149 kinetic trajectories. In contrast, known methods require evaluat-
1150 ing millions of sub-optimal structures to achieve this result. For
1151 the second application, only 46 distinct structures were required
1152 to reproduce the kinetics, whereas known methods required a
1153 sample of 20,000 structures.

1154 Similar to the first part of the result, the second part contains
1155 two chapters. Chapter 4 will briefly introduce the RNA design
1156 problem. It distinguishes the positive from the negative RNA de-
1157 sign problem and reviews the current state of art computational
1158 tools, especially those implementing evolutionary techniques.
1159 The existing tools present challenges when benchmarked on re-
1160 cent datasets such as Eterna100. Another limitation is that most
1161 existing tools do not consider the pseudoknot patterns in their
1162 designing process. In Chapter 5, we proposed an improved evo-
1163 lutionary algorithm inspired by the Lévy flights. Like a Lévy
1164 flight, our tool, aRNAque, implements a Lévy mutation scheme

that allows simultaneous search at all scales over the landscape. New mutations often produce nearby sequences (one-point mutations) but occasionally generate mutant sequences far away in genotype space (macro-mutations). In aRNAque, the number of point mutations distribution at every step is taken to follow a Zipf distribution. The Lévy mutation scheme increases the diversity of designed RNA sequences and reduces the average number of evaluations of the evolutionary algorithm compared to the local search. The overall performance showed improved empirical results compared to existing tools through intensive benchmarks on both pseudoknot (the PseudoBase++ dataset) and pseudoknot-free (the Eterna100 dataset) datasets.

Finally, Chapter 6 presents a general conclusion, a discussion on the results obtained and some promising perspectives. It emphasizes the understanding of the Lévy mutation in the context of RNA design and the connection of our results to evolutionary dynamics.

[June 17, 2022 at 17:55 – 1.0]

1182

Part I

1183

RNA FOLDING

1184

This first part of our thesis provides a literature review
1185 on existing computational tools addressing the pre-
1186 diction of RNA secondary structure, and it presents
1187 our proposed tool RAFFT. Chapter 3 contains figures
1188 and ideas that have previously appeared in our pub-
1189 lication:

1190

- [120] Vaitea Opuu, **Nono SC Merleau**, and Mat-
1191 teo Smerlak (2021). *RAFFT: Efficient prediction of*
1192 *RNA folding pathways using the fast Fourier trans-*
1193 *form.* In: *bioRxiv* (**Submitted**)

[June 17, 2022 at 17:55 – 1.0]

2

1194

1195 INTRODUCTION TO RNA FOLDING

1196 In the previous chapter of the thesis, we provided some motiva-
1197 tions for studying non-coding RNA and introduced the bioin-
1198 formatic concepts of non-coding RNAs. We also highlighted the
1199 relationship between the structure of ncRNAs and their functions.
1200 The functions of ncRNAs and their lengths often distinguish them,
1201 and many ncRNA classes were presented. Identifying the ncRNA
1202 functions is challenging, and their structures largely determine
1203 them. The process of determining the structure of those RNA
1204 molecules is often termed RNA folding. Experimental methods
1205 that determine the secondary structure of such molecules are of-
1206 ten costly. Many computational methods have been developed in
1207 the last decades as alternatives. This chapter provides an overview
1208 of computational methods for predicting RNA secondary struc-
1209 tures. Two techniques will be reviewed: statistical approaches
1210 such as machine learning and score-based methods.

1211 **2.1 STABILITY AND PREDICTION OF RNA SECONDARY STRUC-**
1212 **TURES**

1213 The mapping from RNA sequences to their corresponding sec-
1214 ondary structure defines the folding of RNA molecules. RNA
1215 folding is, therefore, a process by which a linear RNA sequence
1216 acquires a secondary structure through intra-molecular interac-
1217 tions. The nature of those interactions defines the thermodynamic
1218 stability of the secondary structure. The thermodynamic stability
1219 ΔG_σ of a structure σ is the free energy difference with respect to
1220 the completely unfolded state. Therefore, the free energy function
1221 defines the mapping of an RNA sequence to its corresponding free
1222 energy. Most computational methods search for structures that
1223 minimize this free energy function to predict biologically relevant
1224 structures. Structures are decomposed into components called
1225 loops to compute the free energy (See Definition 4). The loop
1226 decomposition allows building the basis of the standard energy
1227 model for RNA secondary structures called the *nearest neighbour*
1228 model [174]. This model associates tabulated free energy values
1229 to loop types and nucleotide compositions; the Turner2004 [106]
1230 and the Turner1999 [107] are the most widely used parameter

sets. The total free energy of a secondary structure is assumed to be a sum over its constituent loops according to the additivity principle [33]. This structure decomposition allows an efficient dynamic programming (DP) algorithm to determine the minimum free energy (MFE) pseudoknot-free structure of a sequence ϕ in the structure space Σ_ϕ .

The DP technique is one of the most widely used score-based methods. The first DP algorithm that finds the structure with the maximum base pairs was proposed by Nussinov, and Jacobson [119]. A few years after, Zucker and Stieger [203] extended Nussinov's algorithm to a more realistic scoring model based on free energy, the NN model. Almost all score-based methods rely on the same DP algorithm, but the decomposition scheme and the scoring model could defer from one to another. When predicting structures with non-canonical base pairs, some other scoring schemes are used, such as nucleotide cyclic motifs score system [27, 122, 154] or equilibrium partition function [156]. Besides the score-based methods, we have the comparative sequence analysis methods, the most computationally accurate for determining the RNA secondary structure [64, 102]. Using the set of homologous structures, This method allows finding base pairs that covary to maintain WC and wobble bases of a given sequence ϕ [65]. The first comparative method predicting a common secondary structure conserved in the given homologous sequence set was developed by Han and Kim in the early nineteenth, and it was based on the comparative phylogenetic analysis. When neglecting the special base pairs and the weak interactions, the running time of both approaches (score-based and comparative analysis) is usually $O(L^3)$ (Where L is the RNA sequence length). Many other comparative analysis methods and variations of score-based methods were also proposed to improve the computational time. More recently, a heuristic method such as LinearFold allows achieving good RNA folding performance in a linear time ($O(L)$).

When pseudoknots are considered, the loop decomposition of a secondary structure and the energy rules break down. Although we can assign reasonable free energies to the helices in a pseudoknot and even to possible coaxial stacking between them, it is impossible to estimate the effects of the new kinds of loops created. Base triples pose an even greater challenge because the exact nature of the triple cannot be predicted in advance, and even if it could, we have no data for assigning free energies. Nevertheless, there are existing techniques that approximate the energies

of pseudoknot loops and allow the dynamic programming technique to tackle the RNA folding with pseudoknots. However, the time complexity still reminds the main problem. Using the DP technique for the pseudoknot structure prediction, the time complexity goes up to $O(L^6)$ for the exact prediction. But for heuristic methods such as IPKnot [142] and Hotknots [129], the running time can be reduced down to $O(L^4)$.

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

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

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

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

1312 Where Σ_ϕ is the set of all possible pseudo-knot free secondary
 1313 structure for the sequence ϕ of lenght L and, $\Delta G(\mathcal{S}, \phi)$ the free
 1314 energy of the structure \mathcal{S} evaluated for the sequence ϕ .

1315 Since each possible structure can be uniquely and recursively
 1316 decomposed into smaller components (or loops) with indepen-
 1317 dent free energy contributions, the DP is best suited for most of
 1318 the following tools presented here.

1319 • **Unfold** [202, 203]: It is the successor of the original `mfold`
 1320 program which was the first realistic implementation of
 1321 the DP for secondary structure predictions with a score
 1322 based on the loop energy parameters and a worse case time
 1323 complexity of $O(L^3)$. The initial version was an improve-
 1324 ment of the simplest DP for secondary structure predic-
 1325 tion known as the *maximum circular matching problem*. The
 1326 authors demonstrated that the loop-based energy model
 1327 is also amenable to the same algorithmic ideas. With Mc-
 1328 Caskill's algorithm [111], for computing the partition func-
 1329 tion of the equilibrium ensemble of RNA molecules, more
 1330 efficient implementations of the initial program with accu-
 1331 rate thermodynamic modelling have been provided. The
 1332 latest implementation is known as `Unfold`.

1333 • **RNAstructure** [109, 130]: The software first appeared in
 1334 1998 as a reimplemention of the program `mfold` with im-
 1335 proved thermodynamic parameters. In its initial version,
 1336 four major changes were made in `mfold`: (1) an improve-
 1337 ment on the methods for forcing base pairs; (2) a filter that
 1338 removed isolated WC or wobble base pairs has been added;
 1339 (3) the energy parameter for interior, internal and hairpin
 1340 loops were incorporated; (4) a new model for coaxial stack-
 1341 ing of helices. It predicts the lowest free energy structure
 1342 and a set of low energy structures. The new implemen-
 1343 tation also provided a user-friendly graphical interface for
 1344 Windows operating system. Subsequently, the first imple-
 1345 mentation was extended to include biomolecular folding; an
 1346 algorithm that finds low free energy structures common to
 1347 two sequences; the partition function algorithm and all free
 1348 energy structures, and the constraints with enzymatic data
 1349 and chemical mapping data. The recent version includes
 1350 the partition function computation for secondary structures
 1351 common to two sequences and can perform stochastic sam-
 1352 pling of common structures [67]. Additionally, it contains

1353 MaxExpect, which finds maximum expected accuracy struc-
1354 tures [100], and a method for removal of pseudoknots, leav-
1355 ing behind the lowest free energy pseudoknot-free struc-
1356 ture.

- 1357 • **RNAfold** [69, 97]: It is one of the most used and efficient fold-
1358 ing tools. It computes the MFE secondary structure using
1359 an efficient DP scheme and backtraces an optimum struc-
1360 ture. It also allows computing the partition function using
1361 McCaskill’s algorithm, the matrix of base pairing probabili-
1362 ties, and the centroid structure. It is part of the ViennaRNA
1363 Package. Since its first version, it aims at suggesting an ef-
1364 ficient implementation of Zucker’s algorithm with more
1365 flexibility on the folding constraints. Many other versions
1366 have been released, including a GPU implementation. The
1367 latest stable release of the ViennaRNA Package is Version
1368 2.5.0.
- 1369 • **LinearFold** [73]: For many decades, the DP techniques
1370 have been the most accurate and fast at predicting pseudoknot-
1371 free structure for short input RNA sequences. But for long
1372 sequences, the prediction remains challenging because of
1373 the computational time and the lack of accurate thermo-
1374 dynamic energy parameters. In contrast to traditional DP
1375 methods which are often bottom-up, LinearFold is a left-
1376 to-right DP. The left-to-right DP consists of scanning the
1377 input RNA sequence ϕ from left to right, maintaining a *stack*
1378 along the way and performing one of the three actions (*push*,
1379 *skip* or *pop*). The *stack* consists of a list of unpaired opening
1380 bracket positions and at each position $j = 1 \dots L$, the three
1381 actions consist respectively of 1) *push*: opening a bracket at
1382 position j , 2) *skip*: unpaired nucleotide at position j and 3)
1383 *pop*: closing the bracket at position j . Initially, LinearFold’s
1384 computational time was similar to the classical DP ($O(L^3)$)
1385 because of the *pop* action that involves three free indices
1386 (i.e. unpaired positions). But using a beam search heuris-
1387 tic, the time complexity was then reduced to $O(Lb \log b)$,
1388 where b is the beam size. The beam search is a popular
1389 heuristic technique used in computational linguistics. This
1390 technique allows keeping only the top b highest-scoring (or
1391 low energy) states for each prefix of the input sequences.

1392 Although the score-based approaches for RNA structure pre-
1393 diction often offer good accuracy and generalization, the non-

1394 availability of the thermodynamic energy parameters for specific
 1395 loops of extended sizes presents the main challenge for predict-
 1396 ing long sequences (i.e. $L \geq 1,000$ nucleotides). Early ML-based
 1397 methods aim to improve the energy parameters by learning the
 1398 underlying folding patterns from a more considerable amount of
 1399 training data. In the next section of this chapter, we will present
 1400 some of the recent improvements in structure prediction using
 1401 ML-based methods.

1402 **2.1.2 ML-based methods**

1403 The ML-based methods for RNA secondary structure prediction
 1404 can generally be classified into three categories according to ML's
 1405 subprocess, i.e., score scheme based on ML, preprocessing and
 1406 postprocessing based on ML, and prediction process on ML. All
 1407 the ML-based methods in these three categories trained their
 1408 models in a supervised way [200].

1409 When using a scoring scheme based on ML, the parameter
 1410 estimation in the scoring scheme is first optimized using an ML
 1411 model. The estimated parameters are then used to evaluate the
 1412 scores of possible conformations. Difference scoring schemes can
 1413 be refined by using that approach: the free energy parameters,
 1414 weights, and probabilities. The free energy parameter-refining
 1415 is the most popular because several thermodynamic parameters
 1416 of the NN model have to be based on a large number of optimal
 1417 melting experiments and the experiments are time and labour-
 1418 consuming. In fact, not all free energy changes in structural el-
 1419 ements can be experimentally measured because of technical
 1420 difficulties. Instead of refining the free energy parameters, some
 1421 ML-based approaches scream through existing data of RNA struc-
 1422 tures to extract weights that consist of different features of RNA
 1423 structure elements. These weights can be used as a scoring func-
 1424 tion for DP techniques. The advantage of such a scoring function
 1425 is that it decouples structure prediction and energy estimation.
 1426 However, learned weights have no explanations because of the
 1427 ML black box.

1428 Another alternative for predicting RNA structures is the stochas-
 1429 tic context-free grammars (SCFG) [39, 88, 89, 133, 138, 189].
 1430 SCFGs allow building grammar rules and induce a joint probabil-
 1431 ity distribution over possible RNA structure for a given sequence
 1432 ϕ . In addition, the SCFG models specify probability parameters
 1433 for each production rule in the grammar, which allow assign-
 1434 ing a probability to each sequence generated by the grammar.

1435 These probability parameters are learned from datasets of RNA
1436 sequences associated with known secondary structures without
1437 carrying any external laboratory experiments [39].

1438 Besides the ML-based methods that focus on refining the fold-
1439 ing parameters, there are preprocessing and post-processing
1440 based on ML [68, 72, 201] and direct predicting process based
1441 on ML [96, 160, 163]. Preprocessing and postprocessing models
1442 allow for choosing the appropriate prediction method or set of
1443 prediction parameter sets and provide a means of determining
1444 the most likely structures among the possible outcomes that are
1445 useful for decision. The preprocessing and postprocessing ML
1446 tools are often based on a support vector machine (SVM).

1447 Finally, it is possible to use ML techniques to predict RNA
1448 secondary structure directly or combine it with other algorithms
1449 in an end-to-end fashion. Below are some of the most used and
1450 recent ML-based tools for RNA secondary structure prediction.

- 1451 • **ContraFold**[36]: Using the so-called probabilistic model,
1452 the conditional log-linear model (CLLM), ContraFold ap-
1453 peared for the first time in early 2006. It was the first prob-
1454 abilistic prediction tool outperforming the existing tools,
1455 including thermodynamic tools such as RNAfold and mfold.
1456 The CLLM is a flexible class of probabilistic models that
1457 generalizes upon SCFGs, using discriminative training and
1458 feature-rich scoring. The tool implements a CLLM incorpo-
1459 rating most of the features found in typical thermodynamic
1460 models allowing the tool to achieve the highest single se-
1461 quence prediction accuracy to date when compared with
1462 the currently available probabilistic models.
- 1463 • **ContextFold** [195]: In contrast to ContraFold, ContextFold
1464 utilizes a weighted approach based on ML. In particular, it
1465 uses a discriminative structured-prediction learning frame-
1466 work combined with an online learning algorithm. ContextFold
1467 uses a large training dataset of RNA sequences annotated
1468 with their corresponding structures to obtain an ML model
1469 made of 70,000 free parameters, which has several orders
1470 of magnitudes compared to traditional models (i.e. ther-
1471 modynamic free energy parameters). At its first apparition,
1472 ContextFold’s model succeeded at the error reduction of
1473 about 50%. Still, some overfitting concerns have been re-
1474 ported when using the tool, especially for predicting struc-
1475 tures with large unpaired regions.

- 1476 • **Mxfold2** [141]: It is one of the most recent ML-based tools
 1477 for predicting the secondary structure of RNA molecules.
 1478 Its particularity is the ML technique used, a Deep Neural
 1479 Network (DNN). it also belongs to the weighted approach
 1480 based on ML since the resulting model of a DNN is a set of
 1481 weight parameters. MxFold2's DNN uses the max-margin
 1482 framework with thermodynamic regularization. It made
 1483 the folding scores predicted by Mxfold2 and the free energy
 1484 calculated by the thermodynamic parameters as close as
 1485 possible. This method has shown robust prediction on both
 1486 sequences and families of natural RNAs, suggesting that
 1487 the weighted ML approaches can compensate for the gaps
 1488 in the thermodynamic parameter approaches.

1489 **2.1.3 Prediction tools for pseudoknotted RNA sequences**

1490 This section introduces a couple of tools for predicting RNA pseu-
 1491 doknotted structures that will be used in the benchmark results
 1492 presented in this thesis. Folding RNA sequences with pseudo-
 1493 knotted interactions is computationally more expensive than a
 1494 pseudoknot-free target. Specifically, the time complexity of the
 1495 pseudoknot-free secondary structure prediction is $O(L^3)$ when
 1496 using dynamic programming approaches such as **RNAfold**, or
 1497 less with heuristic folding methods (e.g. $O(L)$ for **LinearFold**
 1498 and $O(L^2 \log L)$). By contrast, when considering a special class of
 1499 pseudoknots, the time complexity of folding goes up to $O(L^6)$ for
 1500 an exact thermodynamic prediction using a dynamic program-
 1501 ming approach such as [132]. When Using heuristic methods, the
 1502 time complexity slows down to $O(L^4)$ (e.g. tools such as **IPknot**
 1503 and **HotKnots**) or $O(L^3)$ for tool such as **HFold**.

- 1504 • **pKiss** [79]: The program pKiss appears the first time in
 1505 2014 as an updated version of the program pknotsRG[126]
 1506 which is a module of the RNA abstract shapes analysis
 1507 RNAshapes [79]. Initially, the program pknotsRG was built
 1508 for the prediction of some special class of pseudoknots (un-
 1509 knotted structures and H-type pseudoknots). Later on, it
 1510 was extended to predict RNA structures that exhibit kissing
 1511 hairpin motifs in an arbitrarily nested fashion, requiring
 1512 $O(L^4)$ time. In addition to predicting the kissing hairpin mo-
 1513 tifs, pKiss also provides new features such as shape analy-
 1514 sis, computation of probabilities, different folding strategies
 1515 and different dangling base models.

- IPknot [142]: it was first introduced in a paper by Kengo and his collaborators in 2011 as a novel computational tool for predicting RNA secondary structure with pseudoknots using integer programming technique. IPknot uses the maximum expected accuracy (MEA) as a scoring function, and the maximizing expected accuracy problem is solved using integer programming with threshold cut. IPknot decomposes a pseudoknotted structure into a set of pseudoknot-free substructures and approximates a base-pairing probability distribution that considers pseudoknots, leading to the capability of modelling a comprehensive class of pseudoknots and running quite fast. In addition to single sequence analysis, IPknot can also predict the consensus secondary structure with pseudoknots when a multiple sequence alignment is given.
- HotKnots [129]. In contrast to the previously mentioned tools, HotKnots implements a heuristic algorithm based on the simple idea of iteratively forming stable stems. The algorithm explores many alternative secondary structures using a free energy minimization for pseudoknot-free secondary structures. Several other additions of a single substructure are considered for each structure formed at each step, resulting in a tree of candidate structures. The criterion for determining which substructures to add to partially formed structures at successive levels of the tree was also new. Similar to previous algorithms, energetically favourable substructures called “hotspots” are found by a call to Zuker’s algorithm, with the constraint that no base already paired may be in the structure.

1545 2.2 RNA KINETICS

1546 The previous section introduced how secondary structures with
 1547 their thermodynamic properties can be predicted. However, the
 1548 methods used for predictions do not tell us anything about how
 1549 the structures change over time and how they are related to each
 1550 other. In the following section, we will discuss the folding dy-
 1551 namics of RNA molecules.

1552 The folding of RNA molecules is remarkably more complex. It
 1553 is a result of the delicate balance between multiple factors: the
 1554 chain entropy, ion-mediated electrostatic interactions and solva-
 1555 tion effect, base pairing and stacking, and other non-canonical

interactions [22]. It is a dynamic process governed by a constant formation or dissolving of base pairs. In other terms, the RNA molecule navigates its structure space by following a free energy landscape. Here, the free energy landscape is a high-dimensional space of all possible secondary structures (Σ_ϕ) weighted by their free energy ΔG .

As usually done, the kinetics is modelled as a continuous-time Markov chain [99], where populations of structure evolve according to transition rates. In this context, an Arrhenius formulation is commonly used to derive elementary transition from state i to state j ; where $\Delta G_{i \rightarrow j}^\ddagger$ is the activation barrier separating i from j , and $\beta = 1/k_B T$ is the inverse thermal energy (mol/kcal).

$$k_{i \rightarrow j} = k_0 \exp(-\beta \Delta G_{i \rightarrow j}^\ddagger) \quad (2.2)$$

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

1. The base stack model [196–198]: it uses base stacks as elementary kinetic move. A move consists of an addition or a breaking of a base stack with $\Delta G_{i \rightarrow j}^\ddagger$ equal to the change in the entropic free energy $T\Delta S$ and the enthalpy ΔH , respectively.
2. The base pair model [24, 49]: it uses base pair as elementary kinetic steps which gives the finest resolution, but at the cost of computation time. Here $\Delta G_{i \rightarrow j}^\ddagger = \Delta G/2$ where ΔG is the energy change from state i to state j or $\Delta G_{i \rightarrow j}^\ddagger = \Delta G$ for $\Delta G \geq 0$.
3. The helix stem model [76, 104]: the elementary move is the creation or deletion of a helix stem. It provides a coarse-grained description of the dynamics where free energy changes ($\Delta G_{i \rightarrow j}^\ddagger$) due to stem formation guiding the folding process.

The different rate models can lead to different folding pathways. The key factor that distinguishes the different rate models is whether the barrier is determined by $(\Delta H, \Delta S)$ or by ΔG . The $(\Delta H, \Delta S)$ values for different RNA base stacks show well-separated discrete hierarchies, whereas the ΔE values show no such large separation. For two typical base stacks, 5' AU-AU₃' and

1592 5'UC-GA3', the difference $\Delta(\Delta H_{stack}, \Delta S_{stack}) = (7.4 \text{ kcal/mol},$
 1593 $20 \text{ kcal/mol})$ is much larger than the difference $\Delta(\Delta G_{stack}) =$
 1594 1.4 kcal/mol [148]. Because of this fact, different models can give
 1595 different folding kinetics.

1596 Depending on the rate model used, the following master-equation
 1597 describe the population kinetics $p_i(t)$ for the i^{th} state ($i = 1 \dots \Omega$,
 1598 where Ω is the total number of chain conformations).

$$\frac{dp_i(t)}{dt} = \sum_{j \in \Omega} k_{j \rightarrow i} p_j(t) - k_{i \rightarrow j} p_i(t), \quad (2.3)$$

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

$$\frac{d\mathbf{p}(t)}{dt} = \mathbf{M} \cdot \mathbf{p}, \quad (2.4)$$

1599 where $\mathbf{p} = (p_1, \dots, p_\Omega)$ is a column vector representing the frequency of structure at state (i, \dots, Ω) and, \mathbf{M} is the rate matrix defined as:

$$\mathbf{M}_{ij} = \begin{cases} k_{i \rightarrow j}, & \text{if } i \neq j \\ -\sum_{j \neq i} k_{ij}, & \text{if } i = j \end{cases} \quad (2.5)$$

1602 For a given initial folding condition $p_i(0)$, the Equation 2.4 is
 1603 solvable by diagonalizing the rate matrix \mathbf{M} and, the solution is
 1604 the population kinetics $\mathbf{p}(t)$ for $t > 0$ is given by:

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

1605 where $-\lambda$ and \mathbf{n}_m are the m^{th} eigenvalue and eigenvector of
 1606 the rate matrix \mathbf{M} , and C_m is the coefficient that is dependent on
 1607 the initial condition. The eigenvalue spectrum gives the rates of
 1608 the kinetic modes of the system.

1609 Simulating the RNA dynamics using Equation 2.3 has some
 1610 limitations. The solution to the master-equation given by Equation
 1611 2.6 can only give ensemble-average macroscopic kinetics and
 1612 cannot give detailed information about the microscopic pathways
 1613 [199]. Moreover, the number of structures (Ω) increases rapidly
 1614 with the RNA sequence length L . Therefore, the master equation
 1615 is often limited to short RNA sequences. Because of these

1616 limitations, kinetics-cluster methods are alternatively used. The
1617 basic idea of the kinetic-cluster method is to classify the large
1618 structural ensemble into a much-reduced system of clusters (of
1619 macrostates) such that the inter-cluster transitions can represent
1620 the overall kinetics. Although both the master-equation and the
1621 kinetic-cluster methods can predict the macroscopic kinetics, the
1622 kinetic-cluster approach has the unique advantage of providing
1623 direct information on the microscopic pathway statistics from
1624 the inter-cluster transitions [199]. Both approaches are based on
1625 the complete conformational ensemble. An alternative approach,
1626 implemented in kinwalker [58], used the observation that folded
1627 intermediates are generally locally optimal conformations.

1628 Although the above mentioned theoretical models allow simu-
1629 lating the dynamics of RNA folding molecules, they often miss an
1630 essential component, namely, the sequence-dependent confor-
1631 mational statics of the single-stranded coil and loop states. Several
1632 experimental studies have suggested this important component
1633 [14, 117].

1634 In folding experiments, Pan and coworkers observed two kinds
1635 of pathways in the free energy landscape of a natural ribozyme
1636 [121]. Firstly, the investigations revealed fast-folding pathways,
1637 in which a subpopulation of RNAs folded rapidly into the native
1638 state. However, the second population quickly reached metastable
1639 misfolded states, then slowly folded into the native structure. In
1640 some cases, these metastable states are functional. These phe-
1641 nomena are direct consequences of the rugged nature of the
1642 RNA folding landscape [157]. The experiments performed by
1643 Russell and coworkers also revealed the presence of multiple
1644 deep channels separated by high energy barriers on the folding
1645 landscape, leading to fast and slow folding pathways [137]. The
1646 formal description of the above mechanism, called the kinetic
1647 partitioning mechanism, was first introduced by Guo and Thiru-
1648 malai in the context of protein folding [61]. These metastable
1649 conformations constitute competing attraction basins in the free
1650 energy landscape where RNA molecules are temporarily trapped.
1651 However, *in vivo*, folding into the native states can be promoted
1652 by molecular chaperones [20], which means that the active struc-
1653 ture depends on factors other than the sequence. This may raise
1654 some discrepancies when comparing thermodynamic modelling
1655 to actual data. The experimental verification of the rate model is
1656 also a challenge because the microscopic elementary processes
1657 are hidden in the ensemble averages of the measured kinetics.
1658 Many researchers believe that single-molecule experiments may

1659 provide a discerning measure with careful extrapolation to the
1660 force-free case. All atom-simulations with a reliable force field
1661 and sampling method are highly valuable for providing detailed
1662 atomistic configurations for the transition state [22]. Alternatively,
1663 systematic theory-experiment tests as done in [199] for designed
1664 sequences can also provide critical assessment for the different
1665 rate models.

1666 **2.3 CONCLUSION**

1667 In this chapter, we have presented the RNA folding in two main
1668 steps: (1) the prediction of the secondary structure of RNA, which
1669 represents the static part of the folding process; (2) the RNA ki-
1670 netics, which aim at modelling the dynamics of the folding. The
1671 prediction of RNA secondary structure was introduced as an op-
1672 timization problem, and a review of existing methods and tools
1673 was presented. In the next chapter, we will present the first result
1674 of our thesis, which aims at predicting RNA folding pathways
1675 efficiently using the fast Fourier transform. The predicted path-
1676 ways will then allow us to derive a set of energetically suboptimal
1677 structures from which we will model the slow folding process of
1678 RNA molecules.

[June 17, 2022 at 17:55 – 1.0]

3

1679

1680 RAFFT: EFFICIENT PREDICTION OF 1681 FAST-FOLDING PATHWAYS OF RNAs

1682 This chapter introduces a novel heuristic algorithm to predict an
1683 ensemble of metastable RNA secondary structures for a given
1684 sequence ϕ . The algorithm is inspired by the kinetic partitioning
1685 mechanism, by which molecules follow alternative folding
1686 pathways to their native structure, some much faster than others.
1687 Similarly, our algorithm RAFFT generates an ensemble of concurrent
1688 folding pathways ending in multiple metastable structures
1689 for each given sequence. We then use the ensemble structures
1690 as finite ensemble states in which the RNA sequence can be at a
1691 given time, and the energy difference from one state to another is
1692 then used to derive a stem rate model. Therefore, our algorithm
1693 also acts as a folding kinetic ansatz.

1694 **3.1 MATERIAL AND METHODS**

1695 In this section of our work, we describe RAFFT's algorithm and
1696 the kinetics ansatz derived from that.

1697 **3.1.1 RAFFT's algorithm description**

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

$$A \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, U \rightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, C \rightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, G \rightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad (3.1)$$

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

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

For example, $X^A(i) = 1$ if $\phi_i = A$. Next, we create a second copy $\bar{\phi} = (\bar{\phi}_L \dots \bar{\phi}_1)$ for which we reversed the sequence order. Then, each nucleotide of $\bar{\phi}$ is replaced by one of the following unit vectors:

$$\bar{A} \rightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ w_{AU} \end{pmatrix}, \bar{U} \rightarrow \begin{pmatrix} w_{AU} \\ w_{GU} \\ 0 \\ 0 \end{pmatrix}, \bar{C} \rightarrow \begin{pmatrix} 0 \\ 0 \\ w_{GC} \\ 0 \end{pmatrix}, \bar{G} \rightarrow \begin{pmatrix} 0 \\ w_{GC} \\ 0 \\ w_{GU} \end{pmatrix}. \quad (3.3)$$

1698 \bar{A} (respectively $\bar{U}, \bar{U}, \bar{G}$) is the complementary of A (respectively
1699 U, C, G). w_{AU}, w_{GC}, w_{GU} represent the weights associated with
1700 each canonical base pair, and they are chosen empirically. We call
1701 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:

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

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

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

1702 For each $\alpha \in \{A, U, C, G\}$, $X^\alpha(i) \times \bar{X}^\alpha(i+k)$ is non zero if sites
1703 i and $i+k$ can form a base pair, and will have the value of the
1704 chosen weight as described above. If all the weights are set to
1705 1, $\text{cor}(k)$ gives the frequency of base pairs for a positional lag k .
1706 Although the correlation naively requires $O(L^2)$ operations, it
1707 can take advantage of the FFT which reduces its complexity to
1708 $O(L \log(L))$.

1709 Large $\text{cor}(k)$ values between the two copies indicate positional
1710 lags k where the frequency of base pairs is likely to be high. How-
1711 ever, this does not allow to determine the exact stem positions.

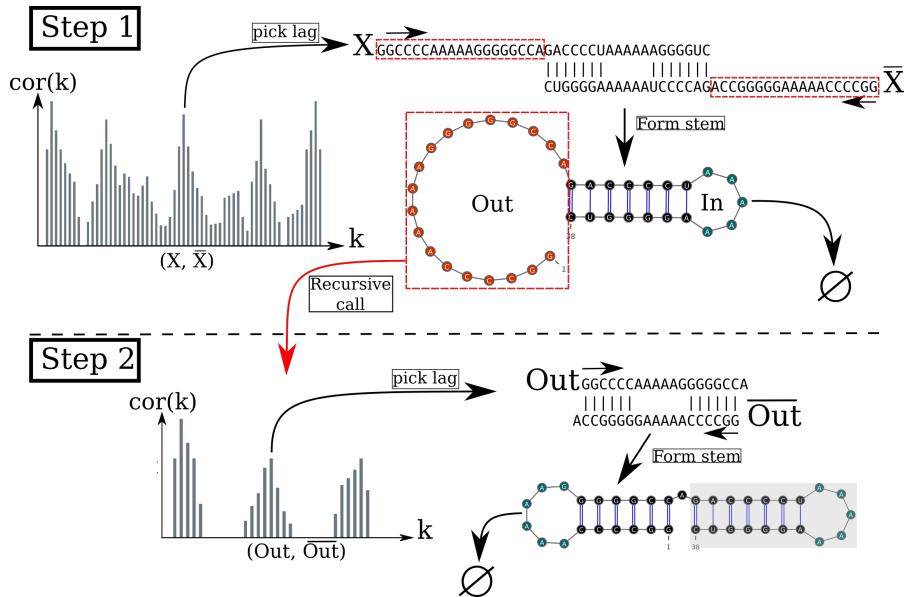


Figure 3.1: Algorithm execution for one example sequence which requires two steps. (Step 1) From the correlation $\text{cor}(k)$, we select one peak which corresponds to a position lag k . Then, we search for the largest stem and form it. Two fragments, “In” (the interior part of the stem) and “Out” (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 $\text{cor}(k)$ 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.

1712 Hence, we use a sliding window strategy to search for the largest
 1713 stem within the positional lag (since the copies are symmetrical,
 1714 we only need to slide over one-half of the positional lag). Once
 1715 the largest stem is identified, we compute the free energy change
 1716 associated with the formation of that stem. Next, we perform the
 1717 same search for the n highest correlation values, which gives us
 1718 n potential stems. Then, we define as the current structure the
 1719 stem with the lowest free energy. Here, free energies were com-
 1720 puted using Turner2004 energy parameters through ViennaRNA
 1721 package API [97].

1722 We are now left with two independent parts, the interior and
 1723 the exterior of the newly formed stem. If the exterior part is com-
 1724 posed of two fragments, they are concatenated into one. Then,
 1725 we apply recursively the same procedure on the two parts inde-
 1726 pendently in a “Breadth-First” fashion to form new consecutive

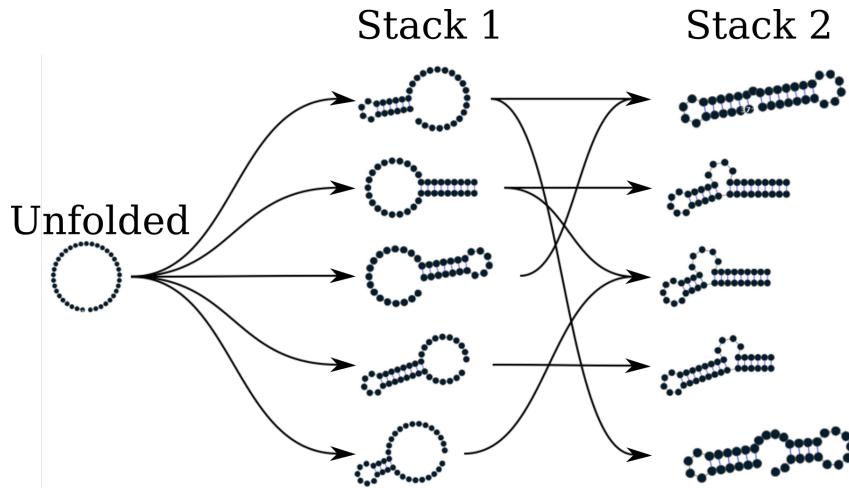


Figure 3.2: Fast folding graph constructed using RAFFT. In this example, the sequence is folded in two steps. The algorithm starts with the unfolded structure on the left. The $N = 5$ best stems are stored in stack 1. From stack 1, multiple stems formation are considered, but only the $N = 5$ best are stored in stack 2. Structures are ordered (from top to bottom) by energy in each stack. All secondary structure visualizations were obtained using VARNA [30].

base pairs. The procedure stops when no base pair formation can improve the energy. When multiple stems can be formed in these independent fragments, we combine all of them and pick the composition with the best overall stability. If too many compositions can be formed, we restrict this to the 10^4 bests in terms of energy. Figure 3.1 shows an example of execution to illustrate the procedure.

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

¹⁷⁴⁷ of a stem or if the two structures are identical. Figure 3.2 shows
¹⁷⁴⁸ an example of a *fast-folding graph* produced by RAFFT for $N = 5$.

¹⁷⁴⁹ 3.1.2 *Kinetic ansatz*

¹⁷⁵⁰ The folding kinetic ansatz used here is derived from the fast-
¹⁷⁵¹ folding graph and allows us to model the slow processes in RNA
¹⁷⁵² folding. As described in Figure 3.2, transitions can occur from
¹⁷⁵³ left to right (and right to left) but not vertically. The fast-folding
¹⁷⁵⁴ graph follows the idea that parallel pathways quickly reach their
¹⁷⁵⁵ endpoints; however, when the endpoints are non-native states,
¹⁷⁵⁶ this ansatz allows slowly folding back into the native state [121].

¹⁷⁵⁷ Using the master-equation (See Equation 2.3), the traditional
¹⁷⁵⁸ kinetic approach often starts by enumerating the whole space (or
¹⁷⁵⁹ a carefully chosen subspace) of structures using RNAsubopt. Next,
¹⁷⁶⁰ this ensemble is divided into local attraction basins separated
¹⁷⁶¹ from one another by energy barriers. This coarsening is usually
¹⁷⁶² done with the tool called barriers. Then, following the Arrhenius
¹⁷⁶³ formulation (See Equation 2.2), one simulates a coarse grained
¹⁷⁶⁴ kinetics between basins.

In contrast to traditional kinetics approaches, the connected structures in the RAFFT's fast-folding graph are not always separated by activation barrier energies. Therefore, we computed the transition rates $k_{i \rightarrow j}$ using the Metropolis [87] formulation defined as follow

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

¹⁷⁶⁵ where $\Delta \Delta G_{i \rightarrow j} = \Delta G_j - \Delta G_i$ is the free energy change between
¹⁷⁶⁶ structure σ_i and σ_j . Here, k_0 is a conversion constant that we set
¹⁷⁶⁷ to 1 for the sake of simplicity and we initialize the population
¹⁷⁶⁸ $p_i(0)$ with only unfolded structures; therefore, the trajectory rep-
¹⁷⁶⁹ resents a complete folding process. The frequency of a structure
¹⁷⁷⁰ σ_i evolves according to the master Equation 2.3.

¹⁷⁷¹ Due to this approximation, we referred to our approach as a
¹⁷⁷² *kinetic ansatz*

¹⁷⁷³ 3.1.3 *Benchmark datasets.*

¹⁷⁷⁴ To build the dataset for the folding task application, we started
¹⁷⁷⁵ from the ArchiveII dataset derived from multiple sources [4, 9,
¹⁷⁷⁶ 17, 28, 31, 57, 62, 63, 108, 139, 144, 155, 158, 159, 183, 204, 205].

1777 We first removed all the structures with pseudoknots, since the
 1778 tools considered here do not handle these loops. Next, using the
 1779 Turner2004 energy parameters, we evaluated the structures' ener-
 1780 gies and removed all the unstable structures: structures with en-
 1781 ergies $\Delta G > 0$. This dataset is composed of 2,698 sequences with
 1782 their corresponding known structures. 240 sequences were found
 1783 multiple times (from 2 to 8 times); 19 of them were mapped to
 1784 different structures. For the sequences that appeared with differ-
 1785 ent structures, we picked the structure with the lowest energy. In
 1786 the end we arrived at a dataset with 2,296 sequences-structures.

1787 For validation of our kinetic ansatz, we used the coronavirus
 1788 frameshifting stimulation element (CFSE) RNA sequence and
 1789 classic bi-stable sequence **GGCCCCUUUUGGGGGCCAGACC-**
 1790 **CCUAAAGGGGUC**.

1791 *3.1.4 Structure prediction protocols*

1792 To evaluate the structure prediction accuracy of the proposed
 1793 method, we compared RAFFT to five recent secondary structure
 1794 pseudoknot-free prediction tools. The five tools include ML-based
 1795 methods (`Mfold2 0.1.1` and `Contrafold`) and score-based meth-
 1796 ods (`RNAfold 2.4.13`, `Linearfold`, and `RNAstructure`). To com-
 1797 pute the MFE structure for the score-based methods, we used the
 1798 default parameters and the Turner2004 set of energy parameters.
 1799 We also computed the ML predictions using the default param-
 1800 eters. Therefore, only one structure prediction per sequence for
 1801 those two methods was used for the statistics.

1802 Two parameters are critical for RAFFT, the number of posi-
 1803 tional lags in which stems are searched, and the number of struc-
 1804 tures stored in the stack. For our computational experiments,
 1805 we searched for stems in the $n = 100$ best positional lags and
 1806 stored $N = 50$ structures. The correlation function $\text{cor}(k)$ which
 1807 allows to choose the positional lags is computed using the weights
 1808 $w_{GC} = 3$, $w_{AU} = 2$, and $w_{GU} = 1$.

1809 To assess the performance of RAFFT, we analyzed the output in
 1810 two different ways. First, we considered only the structure with
 1811 the lowest energy found for each sequence. This procedure allows
 1812 us to assess RAFFT performance in predicting the MFE structure.
 1813 Second, we computed the accuracy of all $N = 50$ structures saved
 1814 in the last stack for each sequence and displayed only the best
 1815 structure in terms of accuracy. As mentioned above, the lowest
 1816 energy structure found may not be the active structure. Therefore,

1817 this second assessment procedure allows us to show whether one
 1818 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:

$$PPV = \frac{TP}{TP + FP}, \quad \text{Sensitivity} = \frac{TP}{TP + FN}, \quad (3.7)$$

1819 where TP, FN, and FP stand respectively for the number of cor-
 1820 rectly predicted base pairs (true positives), the number of base
 1821 pairs not detected (false negatives), and the number of wrongly
 1822 predicted base pairs (false positives). To be consistent with pre-
 1823 vious studies, we computed these metrics using the `scorer` tool
 1824 provided by Matthews *et al.* [105], which also provides a more
 1825 flexible estimate where shifts are allowed.

1826 Further more, we used a Principal Component Analysis (PCA)
 1827 to visualize the loop diversity in the predicted structures for each
 1828 folding tool considered here. To extract the weights associated
 1829 with each structure loop from the dataset, we first converted
 1830 the structures into weighted coarse-grained tree representation
 1831 [149]. In the tree representation, the nodes are generally labelled
 1832 as E (exterior loop), I (interior loop), H (hairpin), B (bulge), S
 1833 (stacks or stem-loop), M (multi-loop) and R (root node). We
 1834 separately extracted the corresponding weights for each node,
 1835 and the weights are summed up and then normalized. Excluding
 1836 the root node, we obtained a table of 6 features and n entries. This
 1837 allows us to compute a 6×6 correlation matrix that we diagonalize
 1838 using the `eigen` routine implemented in the `scipy` package. For
 1839 visual convenience, the structure compositions were projected
 1840 onto the first two Principal Components (PC).

1841 3.2 EXPERIMENTAL RESULTS

1842 3.2.1 RAFFT's run time and scalability

1843 The complexity of RAFFT's algorithm depends on the number
 1844 and size of the stems formed. The main operations performed
 1845 for each stem formed are: (1) the evaluation of the correlation
 1846 function $\text{cor}(k)$, (2) the sliding-window search for stems, and (3)
 1847 the energy evaluation. We based our approximate complexity on

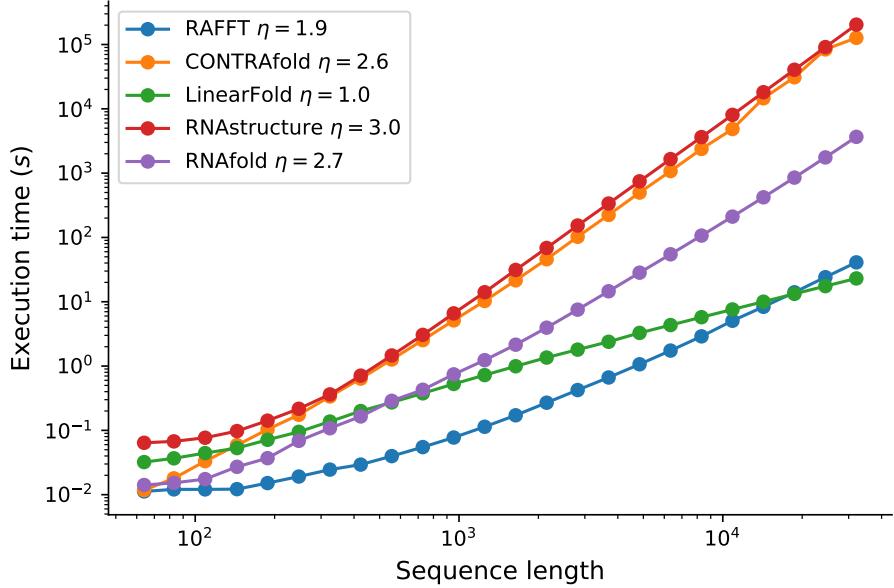


Figure 3.3: **Execution time comparisons.** For samples of 30 sequences per length, we averaged the execution times of five folding tools. The empirical time complexity $O(L^\eta)$ where η is obtained by non-linear regression. RAFFT denotes the naive algorithm where only $N = 1$ structure can be saved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

1848 the correlation evaluation since it is the more computationally
 1849 demanding step; the other operations only contribute a multi-
 1850 pllicative constant at most. The best case is the trivial structure
 1851 composed of one large stem where the algorithm stops after eval-
 1852 uating the correlation on the complete sequence. At the other
 1853 extreme, the worst case is one where at most $L/2$ stems of size 1
 1854 (exactly one base pair peer stems) can be formed. The approxi-
 1855 mate complexity therefore depends on

$$\sum_{i=0}^{L/2} (L - 2i) \log(L - 2i) = O(L^2 \log L) \quad (3.8)$$

1856 We compared RAFFT’s execution time to the classical cubic-time
 1857 algorithms represented by CONTRAfold (Version 2.02), RNAstructure
 1858 (Version 2.0), RNAfold (Version 2.4.13) and the recent improved
 1859 DP tool LinearFold (Version 1.0). Figure 3.3 shows the execution
 1860 time of the RUST implementation of RAFFT and the four above-
 1861 mentioned tools for 30 random generated sequences of various
 1862 lengths. When comparing RAFFT implementation to the standard
 1863 DP tools, the execution time of RAFFT scales slower (with an ex-
 1864 ponent ≈ 2) with the sequence length whereas the standard DP

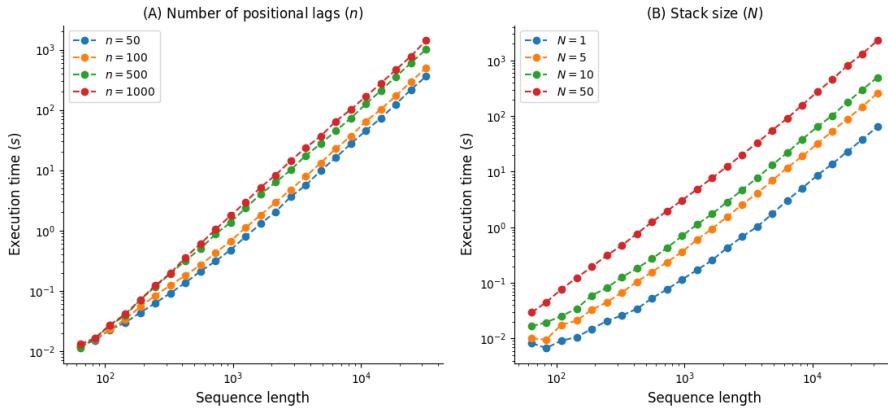


Figure 3.4: Impact of the number of positional lags n and the stack size N on the runtime complexity. For a corresponding length, we generated 30 random sequences, and averaged their execution times. Solid lines indicate the estimated time complexity $O(L^\eta)$ where η is obtained with a non-linear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

1865 execution times are cubic. In contrast, the execution time of the
 1866 improved DP implemented by LinearFold scales linearly with
 1867 the sequence length. Only when considering a stack size of 1,
 1868 that RAFFT execution time is lower than the one of LinearFold
 1869 for sequence of lengths less than $L = 10^4$ (See Appendix A?).

1870 We also analyse the scalability of RAFFT computational time
 1871 with respect to its critical parameters (the number of positional
 1872 lag n and the stack size N). Figure 3.4 shows for both different
 1873 stack sizes and number of positional lags, RAFFT execution time
 1874 against the sequence length. For both stack size and number of
 1875 positional lags, the execution time scales almost with the same
 1876 exponent (≈ 2).

1877 3.2.2 Accuracy of the predicted structural ensemble

1878 We started by analyzing the prediction performances with re-
 1879 spect to sequence lengths: we averaged the performances at fixed
 1880 sequence length. Figure 3.5 shows the performance in predicted
 1881 positive values (PPV) and sensitivity for the five methods. It
 1882 shows that the ML method (Mfold2) consistently outperformed
 1883 RAFFT and the other predictions. When comparing only the MFE
 1884 predictions produced using the DP tools, LinearFold outper-
 1885 formed all other tools (RNAfold and RNAstructure) for both short
 1886 and long sequences. The t -test between the ML and the most used

1887 MFE prediction tool (**RNAfold**) revealed not only a significant
1888 difference (p -value $\approx 10^{-12}$) but also a substantial improvement
1889 of 14.5% in PPV. RAFFT showed performances similar to RNAfold;
1890 but, RAFFT is significantly less accurate (p -value ≈ 0.0002), with
1891 a drastic loss of performance for sequences of length greater than
1892 300 nucleotides (See also Table 3.1).

1893 However, are there relevant structures in the ensemble pre-
1894 dicted by our method? To address this question we retained
1895 the structure with the best score among the 50 recorded struc-
1896 tures per sequence. We obtained an average PPV of 60.0% and
1897 an average sensitivity of 62.8% over all the dataset. The gain in
1898 terms of PPV/sensitivity is especially pronounced for sequences
1899 of length ≤ 200 nucleotides, indicating the presence of biolog-
1900 ically more relevant structures in the predicted ensemble than
1901 the thermodynamically most stable one (PPV was =79.4%, and
1902 sensitivity=81.2%). The average scores are shown in Table 3.1.
1903 We also investigated the relation to the number of bases between
1904 paired bases (base pair spanning), but we found no striking effect,
1905 as already pointed out in one previous study [1].

**Table 3.1: Average performance displayed in terms of PPV and sen-
sitivity.** 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 nu-
cleotides.

	RNAfold	LinearFold	RNAstructure	CONTRAFold	MXfold2	RAFFT	RAFFT*
All sequences							
PPV	55.9	60.6	54.7	58.4	70.4	47.7	60.0
Sensitivity	63.3	58.9	61.5	65.2	77.1	52.8	62.8
Sequences with lengths ≤ 200							
PPV	59.5	63.2	58.2	60.5	76.7	57.9	79.4
Sensitivity	65.5	59.4	63.8	65.9	82.9	63.2	81.2

1906 All methods performed poorly on two groups of sequences:
1907 one group of 80 nucleotides long RNAs, and the second group of
1908 around 200 nucleotides (three examples of such sequences are
1909 shown in the Appendix A3.1). Both groups have large unpaired
1910 regions, which for the first group lead to structures with aver-
1911 age free energies 9.8 kcal/mol according to our dataset. The PCA

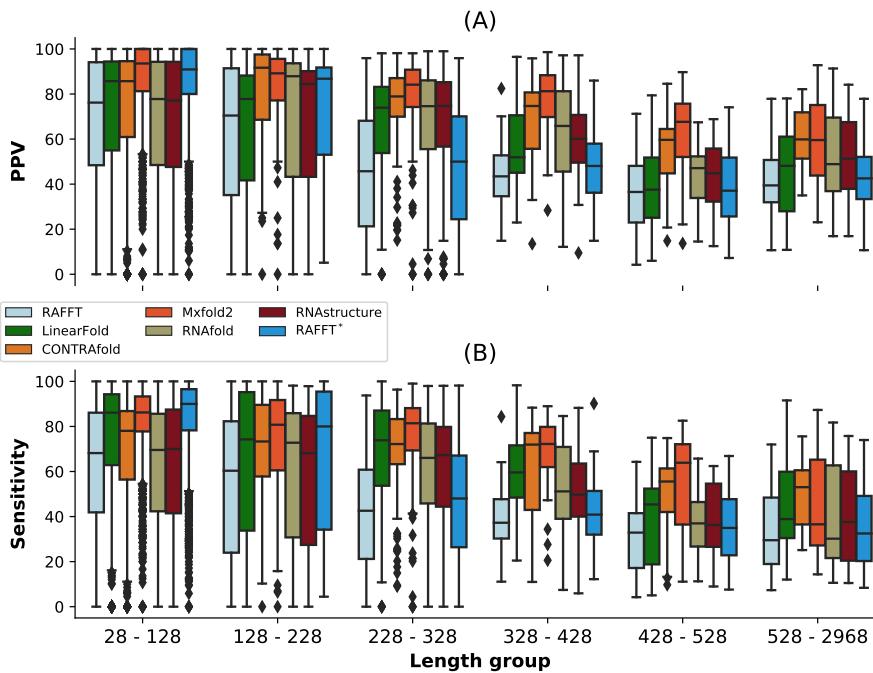


Figure 3.5: RAFFT’s performance on folding task. (A) PPV *vs* sequence length. In the top panel, RAFFT (in light blue) shows the PPV score distributions when for the structure (out of $N = 50$ predictions) with the lowest free energy, whereas RAFFT* (in blue) shows the best PPV score in that ensemble.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

analysis of the native structure space, shown in Figure 3.6, reveals a propensity for interior loops and the presence of large unpaired regions like hairpins or external loops. Figure 3.6 shows the structure space produced by Mfold2, which seems close to the native structure space. In contrast, the structure spaces produced by RAFFT and RNAfold are similar and more diverse.

3.2.3 Applications to the RNA kinetics

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

Figures 3.7A and 3.7B 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

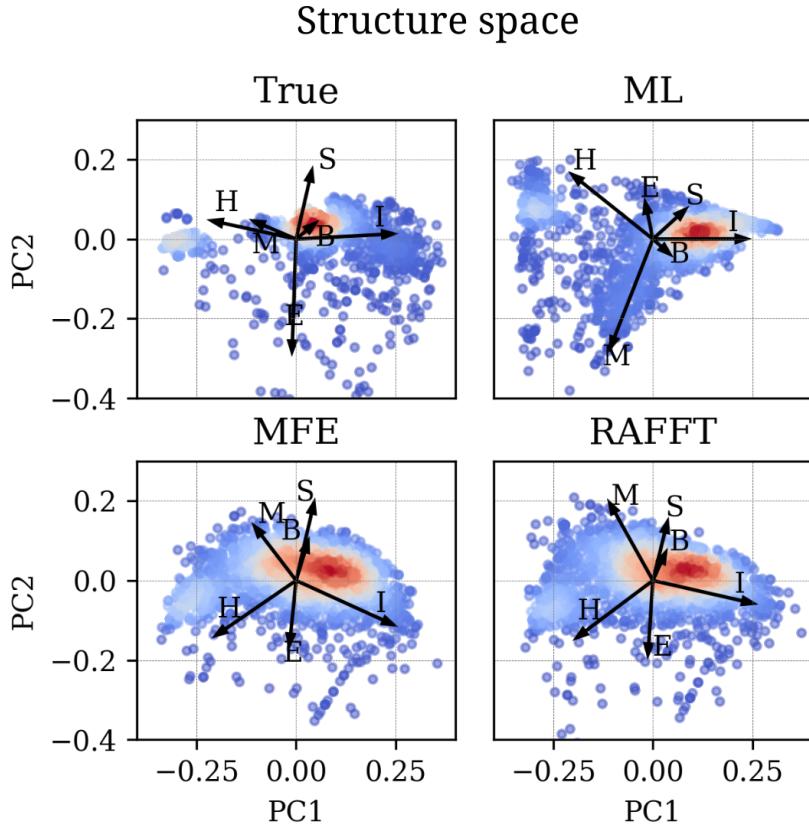


Figure 3.6: **Structure space analysis.** PCA for the predicted structures using RAFFT, RNAfold, MxFold2 compared to the known structures denoted “True”.

their free energies) are stored in a stack. The resulting fast-folding graph consists of 68 distinct structures, each of which is labelled by a number. Among the structures in the graph, 6 were found similar to the native structure (16/19 base pairs differences). The structure labelled “29” in the graph leading to the MFE structure “59” is the 9th in the second stack. When storing less than 9 structures in the stack at each step, we cannot obtain the MFE structure using RAFFT; this is a direct consequence of the greediness of the proposed method. To visualize the energy landscape drawn by RAFFT, we arranged the structures in the fast-folding graph onto a surface according to their base-pair distances; for this we used the multidimensional scaling algorithm implemented in the `scipy` package. Figure 3.7D shows the landscape interpolated with all the structures found; this landscape illustrates the bi-stability of the CFSE, where the native and MFE structures are in distinct regions of the structure space.

From the fast-folding graph produced using RAFFT, the transition rates from one structure in the graph to another are computed

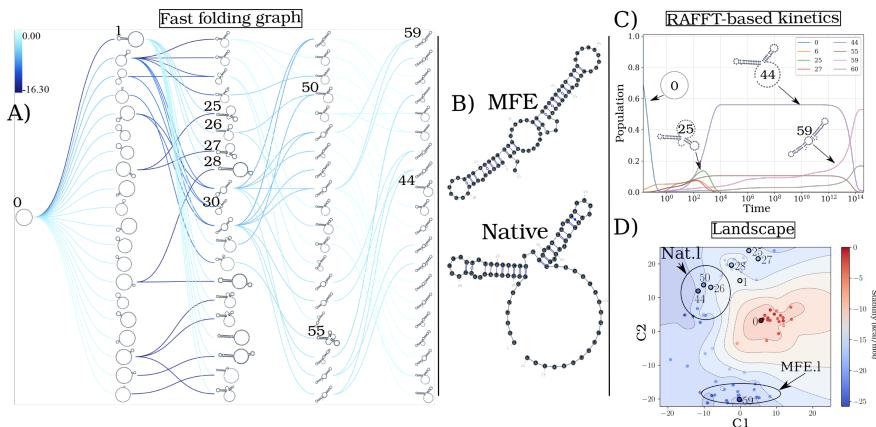


Figure 3.7: Application of the folding kinetic ansatz on CFSE. (A) Fast-folding graph in four steps and $N = 20$ structures stored in a stack at each step. The edges are coloured according to $\Delta\Delta G$. At each step, the structures are ordered by their free energy from top to bottom. The minimum free energy structure found is at the top left of the graph. A unique ID annotates visited structures in the kinetics. For example, “59” is the ID of the MFE structure. (B) MFE (computed with RNAfold) and the native CFSE structure. (C) The change in structure frequencies over time. The simulation starts with the whole population in the open-chain or unfolded structure (ID 0). The native structure (Nat.I) is trapped for a long time before the MFE structure (MFE.I) 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.I) are at the bottom of the figure, while native-like (Nat.I) are at the top.

1946 using the formula given in Eq 3.6. Starting from a population
 1947 of unfolded structure and using the computed transition rates,
 1948 the native of structures is calculated using Eq 2.3. Figure 3.7C
 1949 shows the frequency of each structure; as the frequency of the un-
 1950 folded structure decreases to 0, the frequency of other structures
 1951 increases. Gradually, the structure labelled “44”, which repre-
 1952 sents the CFSE native structure, takes over the population and
 1953 gets trapped for a long time, before the MFE structure (labelled
 1954 “59”) eventually becomes dominant. Even though the fast-folding
 1955 graph does not allow computing energy landscape properties
 1956 (saddle, basin, etc.), the kinetics built on it reveals a high barrier
 1957 separating the two meta-stable structures (MFE and native).

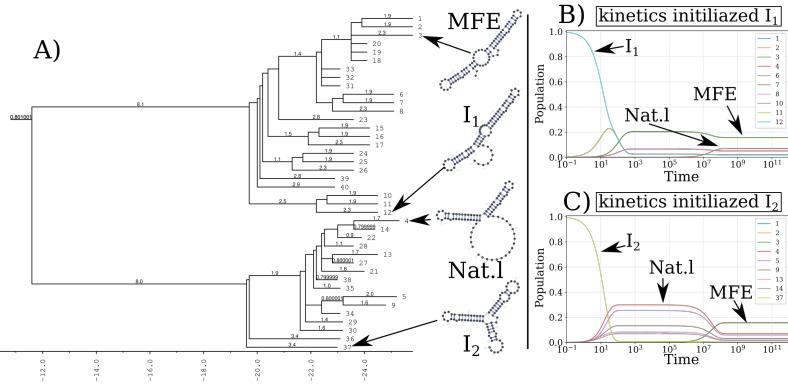


Figure 3.8: Folding kinetics of CFSE using Treekin. A) Barrier tree of the CFSE. From a set of 1.5×10^6 sub-optimal structures, 40 local minima were found, connected through saddle points. The tree shows two alternative structures separated by a high barrier with the global minimum (MFE structure) on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

1958 Our kinetic simulation was then compared to Treekin [51].
 1959 First, we generated 1.5×10^6 sub-optimal structures up to 15 kcal/mol
 1960 above the MFE structure using RNAsubopt [97]. Since the MFE
 1961 is $\Delta G_s = -25.8$ kcal/mol, the unfolded structure could not be
 1962 sampled. Second, the ensemble of structures is coarse-grained
 1963 into 40 competing basins using the tool barriers [51], with the
 1964 connectivity between basins represented as a barrier tree (see
 1965 Figure 3.8A). When using Treekin, the choice of the initial pop-
 1966 ulation is not straightforward. Therefore we resorted to two initial
 1967 structures I_1 and I_2 (see Figure 3.8B and 3.8C, respectively). In
 1968 Figure 3.8B, the trajectories show that only the kinetics initialized
 1969 in the structure I_2 can capture the complete folding dynamics of
 1970 CFSE, in which the two metastable structures are visible. Thus,
 1971 in order to produce a folding kinetics in which the native and
 1972 the MFE structures are visible, the kinetic simulation performed
 1973 using Treekin required a particular initial condition and a barrier
 1974 tree representation of the energy landscape built from a set of
 1975 1.5×10^6 structures. By contrast, using the fast-folding graph pro-
 1976 duced by RAFFT, which consists only of 68 distinct structures, our

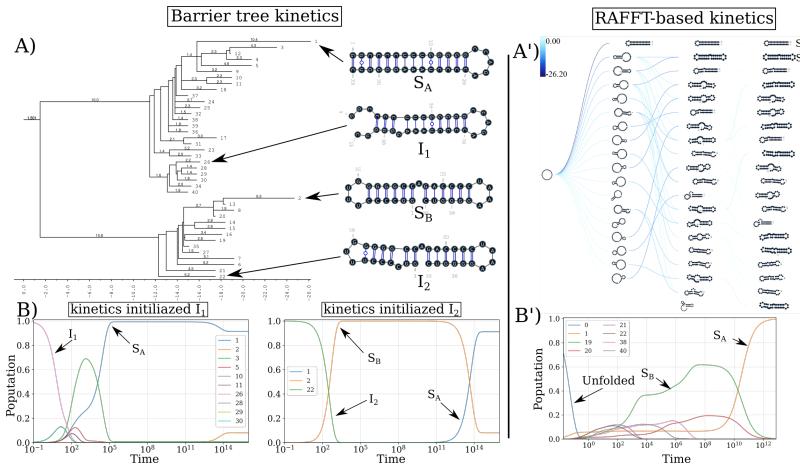


Figure 3.9: RAFFT vs Treekin: folding kinetics of a bi-stable RNA sequence. (A) Barrier tree for the bi-stable example sequence. The local minima and the corresponding barriers are computed from the complete enumeration of the structure space. The bi-stability is visible on the barrier tree through the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

1977 kinetic simulation produces complete folding dynamics starting
1978 from a population of unfolded structure.

1979 As a second illustrative example, we applied both kinetic mod-
1980 els to the classic bi-stable sequence. For Treekin, we first sampled
1981 the whole space of 20×10^3 sub-optimal structures from the un-
1982 folded state to the MFE structure, and from that set, 40 basins
1983 were also computed using barriers. The barrier tree in Figure
1984 3.9 shows the bi-stable landscape, where the two deepest minima
1985 are denoted S_A and S_B . As in the first application, we also chose
1986 two initializations with the structures denoted I_1 and I_2 in Figure
1987 3.9A and 3.9B. Secondly, we simulate the kinetics starting from
1988 the two initial conditions (See Figure 3.9B). When starting from

1989 I_2 , the slow-folding dynamics is visible: S_B first gets kinetically
1990 trapped, and the MFE structure (S_A) only takes over later on.
1991 For our kinetic ansatz, we started by constructing the fast-folding
1992 graph using RAFFT, consisting of only 46 distinct structures. The
1993 resulting kinetics, shown in Figure 3.9B' was found qualitatively
1994 close to the barrier kinetics initialized with structure I_2 . Once
1995 again, with few as 48 structures, our proposed kinetic ansatz can
1996 produce complete folding dynamics starting from a population
1997 of unfolded structure.

1998 3.3 CONCLUSION

1999 We have proposed a method for RNA structure and dynamics
2000 predictions called RAFFT. Our method was inspired by the ex-
2001 perimental observation of parallel fast-folding pathways. We de-
2002 signed an algorithm that produces parallel folding pathways, in
2003 which stems are formed sequentially, to mimic this observation.
2004 Then, to model the slow part of the folding process, we proposed
2005 a kinetic ansatz that exploits the parallel fast-folding pathways
2006 predicted.

2007

Part II

2008

RNA DESIGN

2009

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

2013

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

2014

2015

2016

2017

2018

2019

2020

2021

[June 17, 2022 at 17:55 – 1.0]

4

2022

2023 INTRODUCTION TO RNA DESIGN

2024 The previous chapters demonstrated the implications of non-
2025 coding RNA molecules in varying levels of cellular processes,
2026 from gene expression regulation (miRNAs, piRNAs, lncRNAs)
2027 to RNA maturation (snRNAs, snoRNAs) and protein synthesis
2028 (rRNAs, tRNAs). Knowing that those biological functions are
2029 performed by high dimensional RNA structures, which strongly
2030 depend on their secondary structures, we also provided a compre-
2031 hensive review of computation methods for predicting secondary
2032 structures. Now that we have computational folding tools that
2033 are accurate enough, is it possible to design an RNA molecule
2034 that can accomplish a desired biological function for a given sec-
2035 ondary structure? Answering this question may demand both
2036 experimental and computational efforts. For artificial non-coding
2037 RNAs for which the native RNA sequence is unknown, the es-
2038 sential prerequisite for experimentalists is often a computational
2039 solution to the inverse folding problem. Unlike the folding situa-
2040 tion, the secondary structure is given here, and the goal is to find
2041 one or many RNA sequences that fold into that secondary struc-
2042 ture. This chapter aims to provide the formal background and
2043 biotechnological implications of addressing this problem. Then,
2044 it gives a brief literature review of the existing computational
2045 methods.

2046 **4.1 RNA INVERSE FOLDING AND BIOTECHNOLOGICAL IMPLICA-**
2047 **TIONS**

2048 In modern biotechnology, we often seek to reproduce the natural
2049 ability of the cells to control gene expressions using a variety of
2050 nucleic acids and proteins. These natural cellular abilities result
2051 from networks of regulatory molecules such as ncRNAs that dy-
2052 namically regulate the expression of specific genes in response
2053 to environmental signals. Therefore, the ability to engineer bio-
2054 logical systems is directly related to controlling gene expression.
2055 The increasing number of examples of natural regulator ncR-
2056 NAs has opened doors to many emerging subfields such as RNA
2057 synthetic biology [21, 75] and RNA nanostructure [60, 78]. Re-
2058 searchers have engineered RNA molecules with new biological

2059 functions, inspired by this natural versatility. Synthetic biology
 2060 has also made significant progress in developing versatile and
 2061 programmable genetic regulators that precisely control gene ex-
 2062 pressions in the last decades. Three general approaches are taken
 2063 to engineer new functional RNAs: harvesting from nature, com-
 2064 putational design and molecular evolution. We are interested
 2065 here in computational design methods.

2066 In most cases, designing a functional RNA goes beyond com-
 2067 putationally generating a set of RNA sequences that fold into
 2068 a given structure. Successful design methods include computa-
 2069 tional and experimental, predictive and analytical techniques.
 2070 However, computational tools addressing the inverse folding
 2071 problem often provide some guidance and rationalities through
 2072 the design process. For example, Steffen Mueller and his col-
 2073 laborators [116] suggested a systematic, rational approach, Syn-
 2074 thetic Attenuated Virus Engineering (SAVE), to develop new,
 2075 productive live attenuated influenza virus vaccine candidates
 2076 using computer-aided rational design. In addition, Eckart Binde-
 2077 wald et al. [12] used computational tools for solving inverse RNA
 2078 folding in the design of nanostructures, including pseudoknots.
 2079 And in designing several ncRNAs with a successful synthetic
 2080 such as ribozymes [38], riboswitches [48, 179].

2081 4.2 THE POSITIVE AND NEGATIVE DESIGN.

2082 We often find two types of design problems for RNAs in the
 2083 literature: a positive design and a negative one. The negative
 2084 structural design of RNAs, also called the inverse RNA folding
 2085 problem, aims to find one or many RNA sequences that fold into
 2086 a given target RNA secondary structure while avoiding alterna-
 2087 tive folds of similar quality for the chosen energy model ΔG . In
 2088 other terms, it is an optimization problem where a target RNA
 2089 secondary structure \mathcal{S}^* of length L is given, and the goal is to de-
 2090 termine an RNA sequence ϕ of length L such that $\forall \mathcal{S} \neq \mathcal{S}^* \in \Sigma_\phi, \Delta G(\phi, \mathcal{S}) > \Delta G(\phi, \mathcal{S}^*)$.

2092 This problem is NP-hard even in a simple energy model [13],
 2093 and we can not provide a parameterized algorithm that solves it
 2094 in a polynomial time.

2095 In contrast, a positive design problem consists of optimizing
 2096 affinity towards a given target secondary structure. In another
 2097 terms, the objective is to find a sequence $\phi \in \{A, U, C, G\}^L$ such
 2098 that $\mathcal{S}^* = \mathcal{S}^{MFE}(\phi) = \arg \min_{\mathcal{S} \in \Sigma_\phi} \Delta G(\phi, \mathcal{S})$ (i.e. the sequence
 2099 ϕ should have as MFE structure of its ensemble Σ_ϕ the target

structure \mathcal{S}^*). The positive design is computationally solvable exactly in a polynomial time [50].

Both negative and positive designs are considered in this work, and the main difference often depends on the objective function used. In addition, it has been recently shown that the proportion of designable secondary structures decreases exponentially with L for various popular combinations of energy models and design objectives [191].

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

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

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

- A simple distance from the target structure: in the simplest setting, the objective function of an RNA sequence ϕ defines the distance between \mathcal{S}^* and the current MFE structure $\mathcal{S}^{MFE}(\phi)$. It often requires only the MFE structure's computation, hence being computationally fast. There are many variants of this distance measure: base-pair distance, ham-

ming or string edit distance, tree-edit distance and energy distance. For a formal definition of each of those distances, see Section 1.4. This objective function was used in the earliest tools such as RNAinverse [69] but also in many others since then [5, 18, 55].

- A negative design objective function: in contrast to the above mentioned objective functions (often considered when performing a positive design), we consider the whole structural ensemble when computing the fitness of an RNA sequence ϕ . In most cases, it is preferable also to consider negative design goals, which allows for avoiding alternative structures of similar quality to the target structure. The notion of defect often terms the avoidance of alternative structures. Negative RNA design methods usually consider one of the three following defects: (1) the *suboptimal defect* [35, 50, 69, 193] which defines the energy distance to the first suboptimal (2) the *probability defect* [69, 193] which defines the probability that the sequence ϕ folds into any other structure than the target structure S^* and (3) the *ensemble defect* [193] which corresponds to the average number of incorrectly paired nucleotides at equilibrium calculated over the structure ensemble of ϕ , Σ_ϕ .
- Multi-objective optimization: in some designing cases where more than one goal is specified, it is necessary to formulate an objective function for each goal. That results in a multi-objective optimization problem. The solutions to such a problem are all optimal for at least one objective function and thus arranged on the so-called Pareto optimal front. This approach has already been used in several RNA design tools such as Modena [165, 166] and in [125].
- Bistable and multi-stable riboswitches objective functions: In some designing cases, especially for riboswitches, it is possible to specify more than one desired target structure, including the energy differences between them, the barrier heights and the kinetic properties. Following the same idea, Flamm et al. introduced an objective function that enables designing RNA molecules to adopt two distinct structures [50]. This bistable objective function contains two terms. The first term increases the probability of both structures in the ensemble, and the second specifies the desired energy difference between both states. It is also possible to vary

the states' temperature to gain a bistable thermoswitch. The same idea has therefore been expanded to an objective function for designing RNA molecules that can adopt more than two structures, including extension for multi-structure energy barrier calculations [125, 153]. Frnakenstein [101] also utilises such objective function for multi-target design.

- Mutational robustness and neutrality: In addition to the above-mentioned objective functions, objective functions aim to measure the mutual neutrality of the sequence concerning the target structure [153]. When using such an objective function, the sequences are optimized so that the fraction of one-mutant neighbours to the original structure is as significant as possible. This allows for perfectly preserving the structure when mutations are introduced. We often talk of a mutational robustness optimization [6].

4.4 A REVIEW ON EXISTING INVERSE RNA FOLDING TOOLS.

Several methods or algorithms addressing this problem have been proposed in the literature. The existing techniques can be classified into two categories: one for the pseudoknot-free structure design and another for the pseudoknotted RNA structure design. This section gives a short description of some of the existing tools, especially those used in the benchmark results of the thesis.

4.4.1 Pseudoknot-free RNA inverse folding tools

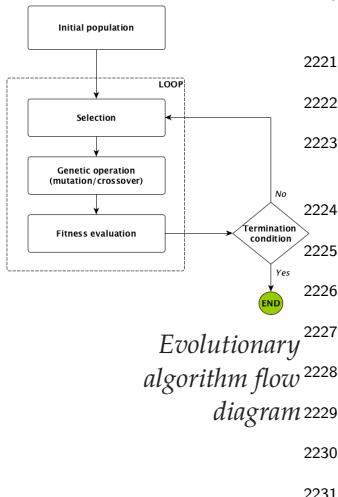
Due to the complexity of the RNA design, most of the existing tools perform a stochastic search optimization where initial potential solutions are generated and refined over a finite number of iterations or generations [40, 43, 44, 123, 164].

4.4.1.1 Evolutionary algorithms and RNA inverse problems

Since the genetic algorithm (or more generally evolutionary algorithm) was proposed by John Holland [71] in the early 1970s, it has emerged as a popular search heuristic and found application in many disciplines that deal with complex landscape optimization problems.

Among the existing tools dealing with the RNA inverse problem, both ERD [43, 44] and MODENA [165] are evolutionary algo-

2217 rithms (EA) but implementing different strategies. In general, an
 2218 evolutionary search algorithm on any fitness landscape consists
 2219 of three main parts, which in the context of RNA inverse folding
 2220 are as follows:



- 2221 • Initialization: generating a random initial population of
 2222 RNA sequences compatible with the given target secondary
 2223 structure.
- 2224 • Evaluation and selection: evaluating a population of RNA
 2225 sequences consists of two steps: 1) fold each sequence into
 2226 a secondary structure and assign it a weight based on its
 2227 similarity to the target structure. 2) select a weighted ran-
 2228 dom sample with replacement from the current population
 2229 to generate a new population. A detailed description of
 2230 the objective function used in our proposed tool aRNAque is
 2231 provided in the next chapter.
- 2232 • Mutation (or move) operation: define a set of rules or steps
 2233 used to produce new sequences from the selected or ini-
 2234 tial ones. This component is elaborated further in the next
 2235 chapter.

2236 MODENA uses a multi-objective function that measures the stabil-
 2237 ity of the folded sequence and its similarities to the target. It starts
 2238 from a population of randomly generated sequences, and the ob-
 2239 jective is optimized through tournament selection and random
 2240 mutation at non-closing loop positions.

2241 In contrast, ERD starts by decomposing the target structure
 2242 into loops and independently uses an evolutionary algorithm
 2243 to minimize each constituent's energy. It was first developed in
 2244 2014 [44], and one year after, an updated version was released
 2245 [43]. The main lines of ERD are:

- 2246 1. Pool reconstruction: using a collection of RNA sequences
 2247 (STRAN database) similar to the natural ones, a pool of
 2248 sequences is constructed for their length by successively
 2249 finding the corresponding structure using RNAfold, decom-
 2250 posing the structure in sub-components, and finally, the
 2251 corresponding sub-sequences of the same size are gathered
 2252 to form a pool.
- 2253 2. Hierarchical decomposition of the target structure into loops:
 2254 using the idea that any secondary structure can be uniquely
 2255 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.

- 2256 3. Sequence initialization: after decomposing the target struc-
2257 ture into sub-components, for each sub-component, a ran-
2260 dom sub-sequence is chosen from the pool, and the initial
2261 sequence is a combination of those sub-sequences;
- 2262 4. Evolutionary optimization of the sub-sequences: an EA al-
2263 gorithm is performed on each sub-component to improve
2264 the initial sequence. The outcome sub-sequences are com-
2265 bined to form a newer sequence that will replace the initial
2266 one. Iteratively the evolutionary algorithm is performed on
2267 the updated sequence until the combined sequence folds
2268 into the target or in a failure case when the stopping con-
2269 dition is satisfied. Two evolutionary operators are imple-
2270 mented here, a mutation that consists of replacing a sub-
2271 sequence corresponding to a sub-component with a new
2272 random one from the pool for the same length, and a se-
2273 lection which consists of choosing from a population of
2274 15 RNA sequences or sub-sequences, three best sequences
2275 with respect to their free energy and adding them to the
2276 best from the previous generation, three best ones with re-
2277 spect to the Hamming distance from the target are therefore
2278 chosen. The next-generation population is then obtained
2279 by generating five new sequences for each of the three best
2280 sequences.

2281 4.4.1.2 Lévy flights and evolutionary algorithms

2282 In this section, we define concepts such as Lévy flights and pro-
2283 vide a brief review of its implications and applications to opti-
2284 mization techniques such as evolutionary algorithms.

2285 In its classical setting, evolutionary algorithms are guided by
2286 local (or one-point mutations) mutations. Although a local search
2287 can efficiently discover optima in a simple landscape, more com-
2288 plex landscapes pose challenges to designing evolutionary al-
2289 gorithms that rely solely on local search. This is especially true
2290 on a landscape with high neutrality where local search may be
2291 inefficient or risk getting stuck on a plateau (or local optimum).
2292 To avoid this pitfall, many practitioners suggested EA that imple-
2293 ments a mutation scheme inspired by Lévy flights (called Lévy
2294 mutation).

2295 Lévy flights are random walks with a Lévy (or any heavy-
 2296 tailed) step size distribution. The concept originates in the work
 2297 of Mandelbrot on the fluctuation of commodities prices in the
 2298 1960s [103] but has since found many more physical applications
 2299 [152]. The term "Lévy flight" was also coined by Mandelbrot, who
 2300 used one specific distribution of step sizes (the Lévy distribution,
 2301 named after the French mathematician Paul Lévy). Lévy flights
 2302 also play a key role in animal foraging, perhaps because they
 2303 provide an optimal balance between exploration and exploitation
 2304 [81, 177]. For a recent review of applications of Lévy flights in
 2305 biology from the molecular to the ecological scale, [131].

2306 Similar to a Lévy flight, a Lévy mutation scheme allows simul-
 2307 taneous search at all scales over the landscape. New mutations
 2308 most often produce nearby sequences (one-point mutations), but
 2309 occasionally generate mutant sequences which are far away in
 2310 genotype space (macro-mutations). In this work, the distribution
 2311 of the number of point mutations at every step is taken to follow
 2312 a Zipf distribution [118].

2313 Earlier works have applied similar ideas in genetic program-
 2314 ming [29], and in differential evolutionary algorithms [150]. This
 2315 motivated us to investigate a possible benefit of a Lévy flight in
 2316 the design of RNA sequences in the next chapter.

2317 4.4.1.3 Tools implementing non-EA strategies.

2318 Several tools dealing with the RNA folding problem implement
 2319 different strategies from the population-based or evolutionary
 2320 algorithm approaches.

2321 `sentRNA` [151] is a computational agent that uses a set of infor-
 2322 mation and strategies collected from the EteRNA game players
 2323 to train a neural network model. The neural network assigns an
 2324 identity of A, U, C, or G to each position in the given target, a
 2325 featured representation of its local environment. The featured
 2326 representation combines information about its bonding partner,
 2327 nearest neighbours, and long-range features. While the bonding
 2328 partner and nearest neighbour information are provided to the
 2329 agent by default, long-range features are learned through the
 2330 training data. For each target structure, the long-range features
 2331 refer to the important position j relative to i that the agent should
 2332 know about when deciding what nucleotide to assign to i . These
 2333 are defined by two values: the Cartesian distance and the angle in
 2334 radians. Those two values are computed for each position (i, j) us-
 2335 ing a mutual information metric over the player solution dataset.

2336 Therefore, the result is a list of long-range features for a given
2337 target structure. A subset of long features is selected from this
2338 list and used to define a model for the neural network model's
2339 training, validation, and testing. In addition to the neural network
2340 model, sentRNA also implements a refinement algorithm on the
2341 unsuccessful design. The refinement algorithm is an adaptive
2342 walk that starts from the predicted sequence and uses a set of
2343 random mutations that allow improving the neural network solu-
2344 tion. Alternatively, EternaBrain [91] implement a convolutional
2345 network model trained on a huge EteRNA moves-select reposi-
2346 tory of 30,477 moves from the top 72 players; and LeaRNA [136]
2347 uses deep reinforcement learning to train a policy network to
2348 sequentially design an entire RNA sequence given a specified
2349 target structure.

2350 NEMO [123] is a recently developed tool combining a Nested
2351 Monte Carlo Search (NMCS) technique with domain-specific
2352 knowledge to create a novel algorithm. The underlying idea is to
2353 start with an input pattern sequence of N's of the same length as
2354 the targeted structure. First, it uses the standard NMCS to sam-
2355 ple sequence solutions acting on N's only. A sequence candidate
2356 is selected from the sample; then folded into an MFE structure.
2357 When the MFE structure does not match the target, some subset
2358 mutations are performed, and a set of random mutated positions
2359 are picked to generate a new input pattern sequence. The new
2360 input pattern will allow sampling acting on N's only using the
2361 same standard NMCS. This procedure is then repeated several
2362 times until the MFE structure matches the targeted structure or
2363 not in the unsuccessful cases. The statistical results show that
2364 NEMO surpasses all the existing tools on the EteRNA100 benchmark
2365 datasets by solving $\approx 95\%$ of the targets using the Turner1999 en-
2366 ergy parameter sets. Using a similar technique, RNAinverse[98],
2367 one of the oldest inverse folding tools included in the ViennaRNA
2368 package, uses an adaptive random walk to minimize base-pair dis-
2369 tance. The distance is computed by comparing the MFE structure
2370 of the mutated sequence with the target structure. In addition,
2371 RNAinverse allows for designing more probable sequences using
2372 the partition function optimization. The latter allows for more
2373 stable designed sequences that mostly fold into MFE structures
2374 different from the target structure. On an attempt to improve
2375 RNAinverse, many other tools have been suggested INFO-RNA [18],
2376 RNA-SSD [5] and DSS-Opt [110]. The most recent tools also include
2377 RNAPOND [192] and MaiRNAiFold [114].

2378 antaRNA [86] is also a recent program available since 2015, and
 2379 it provides a web server for friendly usability. It utilizes an *ant-*
 2380 *colony* optimization, in which an initial sequence is generated via
 2381 a weighted random search, and the *fitness* of that sequence is
 2382 then used to refine the weights and improve subsequences over
 2383 generations. It provides many other interesting features, such
 2384 as the sequence and target GC-content constraints. It also pro-
 2385 vides a fast python script that includes the options from the web
 2386 server presented through a command line. Other tools also pro-
 2387 vide this dual advantage but implement different optimization
 2388 techniques. NUPACK:design [194] uses a tree decomposition tech-
 2389 nique and the ensemble defect as objective function to design
 2390 qualitatively good sequences. incaRNAbinv [41] is a program for
 2391 fragment-based RNA design. incaRNAbinv's web server com-
 2392 bines two complementary methodologies: IncaRNATION [128] and
 2393 RNAbinv [185]. IncaRNATION generates a GC-weighted partition
 2394 function for the target structure, and then adaptively samples se-
 2395 quences from it to match the desired GC-content. RNAiFold [56]
 2396 employs constraint programming that exhaustively searches over
 2397 all possible sequences compatible with a given target. RNAiFold
 2398 [56] has the particularity of designing synthetic functional RNA
 2399 molecules.

2400 4.4.2 Pseudoknotted RNA inverse folding tools

2401 Designing RNA sequences for pseudoknotted targets is compu-
 2402 tionally more expensive than pseudoknot-free targets. For that
 2403 reason, many of the studies addressing the inverse folding of RNA
 2404 considered only pseudoknot-free secondary structures. There are,
 2405 however, some exceptions: MCTS -RNA [190], antaRNA[86], Modena
 2406 and Inv[55]. The computation tool presented in the result chapter
 2407 of our work also considers pseudoknots.

2408 Inv was one of the first inverse folding tools handling pseudo-
 2409 knotted RNA target structures, but it was restricted to a specific
 2410 type of pseudoknot pattern called 3-crossing nonplanar pseudo-
 2411 knots.

2412 More recently, MCTS -RNA's authors suggested a new technique
 2413 that deals with a broader type of pseudoknots. It uses a Monte
 2414 Carlo tree search (MCTS) technique which has recently shown
 2415 exceptional performance in Computer Go. The MCTS allows
 2416 initialising a set of RNA sequence solutions in MCTS -RNA and
 2417 the solutions are further improved through local updates at the
 2418 nucleotide positions.

Another approaches (Modena, antaRNA) implements different strategies one which is a multi-objective ant-colony optimisation and the another one which is a multi-objective evolutionary algorithm. Although the first versions were implemented for pseudoknot-free structure [86, 164], they have since been extended to support pseudoknotted RNAs [85, 165].

MCTS-RNA uses pkiss as folding tool whereas the other tools (antaRNA and Modena) support a broader range of folding tools such as HotKnots or IPKnot.

4.5 BENCHMARKING THE INVERSE FOLDING TOOLS

The validation of the designed RNA sequences using computational methods often requires biological experiments. Because of the high cost of experimental techniques, most investigators limit their guarantee to using benchmark datasets [23] in general. For pseudoknot-free design tools, two benchmark datasets are mostly used in the literature—(i) RFAM¹: a collection of RNA families, each represented by multiple sequence alignments, consensus secondary structures and covariance models—(ii) Eterna100 [3]: a collection of hundred RNA secondary structures extracted from the EteRNA Puzzle game². For RNA inverse tools that support pseudoknots, the PseudoBase++[167] dataset is often considered.

The Eterna100 dataset [92] 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 [173]. Out of the 100 target secondary structures, 19 turned out to be unsolvable using the version of ViennaRNA 2.4.14 (which relays on the Turner2004 [106]). Subsequently, an Eterna100-V2 [92] was released in which the 19 targets were slightly modified to be solvable using ViennaRNA 2.4.14 and any version that supports the Turner2004 energy parameters. The main difference between the two dataset relay on the energy parameters used to generate the data.

The non-EteRNA (a subset of the RFAM) dataset in a set of 63 experimentally synthesized targets that Garcia-Martin et al. [56] recently used to benchmark a set of ten inverse folding algorithms, which from our knowledge, is the most recent and comprehensive

¹ The Rfam database <https://rfam.xfam.org/>

² The EteRNA game <https://eternagame.org/>

2457 benchmark of current state-of-the-art methods. The dataset is
 2458 collected from 3 sources: the first dataset called **dataset A** which
 2459 contains 29 targets collected from RFAM and also used in [43,
 2460 164] and the second called **dataset B** is a collection of 24 targets
 2461 used in [43] and added to that the 10 structures used in [151].

2462 The PseudoBase++ is a set of 266 pseudoknotted RNA struc-
 2463 tures used to benchmark Modena. It was initially 342 RNA sec-
 2464 ondary structures, but because of the redundancy and the non-
 2465 canonical base pairs 76 structures were excluded. To group the
 2466 dataset with respect to the pseudoknot motifs, we used the test
 2467 data from antaRNA's paper. The test data contains 249 grouped
 2468 into four categories: 209 hairpin pseudoknots (H), 29 bulge pseu-
 2469 doknots (B), 8 complex hairpin pseudoknots (cH) and 3 kissing
 2470 hairpin pseudoknots (K). Out of the 266 structures, only 185
 2471 (with 150 H-type, 3 K-type, 25 B-type and 7 cH-type) structures
 2472 were included in the test data. So for that reason, we have used
 2473 only 185 target structures for the pseudoknot motif performance
 2474 comparison and the 266 structures for the different target lengths
 2475 performance comparison.

2476 When the benchmark datasets rely on a particular energy pa-
 2477 rameter set, the performance of a given inverse RNA folding tool
 2478 evaluated on these datasets will also be related to the choice of
 2479 the RNA folding tool's energy parameter set. If the benchmark
 2480 datasets do not rely on a particular energy parameter set, the ro-
 2481 bustness of the inverse RNA tool will be its capability to perform
 2482 well on different energy parameter sets.

2483 4.6 CONCLUSION

2484 In summary, the RNA inverse folding problem is still computa-
 2485 tionally challenging. It finds applications in RNA synthetics, RNA
 2486 nanostructure design, and emerging fields, e.g. bioengineering
 2487 and new biotechnology innovations. We introduced in this chap-
 2488 ter a rich literature review of existing computational methods that
 2489 addressed this problem. The existing approaches have some ad-
 2490 vantages and disadvantages, depending on the techniques imple-
 2491 mented. NUPACK for example—despite its well-defined objective
 2492 function—still has difficulty designing sequences for large tar-
 2493 gets and most of the EterNA100 targets. In contrast, ERD because
 2494 of its powerful decomposition method, which allows dealing
 2495 quickly with large targets (On RFAM 1.0 with target's length be-
 2496 tween 400–1400) but still a big challenge to solve more than 65% of
 2497 the EterNA100-V2 using the Turner2004 energy parameter sets.

2498 On another side, NEM0, one of the most recent tools, can solve
2499 more than 90% of the EteRNA100-V1 dataset using an old version
2500 of ViennaRNA package, which is based on Turner1999 energy pa-
2501 rameter sets [173]. The sentRNA's machine learning model also
2502 relied on the same old version of ViennaRNA package and, by
2503 adding a refinement on the machine learning model, sentRNA
2504 solves 78% of EteRNA100. Without this refinement, sentRNA can
2505 only solve 48% of EteRNA100's targets, which can represents an-
2506 other limitation. For the EAs ERD and MODENA, none of them can
2507 solve more than 65% of EteRNA100 using the Turner2004 energy
2508 parameter sets. In the next chapter, we will introduce a simple
2509 evolutionary algorithm called aRNAque that implements a Lévy
2510 mutation and allows significant improvements to the existing
2511 tools.

[June 17, 2022 at 17:55 – 1.0]

2512

2513 AN EVOLUTIONARY ALGORITHM FOR INVERSE
2514 FOLDING INSPIRED BY LÉVY FLIGHTS.

2515 In the previous chapter of our work, we prevented the RNA de-
2516 sign as an optimization problem and provided a significant liter-
2517 ature review on the existing tools addressing that problem. We
2518 highlighted some limitations of the existing tools, particularly the
2519 ones implementing evolutionary algorithms. One of the evolu-
2520 tionary algorithms' main challenges is to avoid deception, which
2521 is the fast convergence to a local optimum. Most evolutionary
2522 algorithms' early convergence to a local optimum is due to the
2523 local search implementation, which is the consequence of the
2524 local mutation scheme.

2525 An alternative mutation scheme to the classical local search
2526 is the Lévy mutation to avoid this pitfall. We propose an evo-
2527 lutionary algorithm implementing a similar Lévy mutation in
2528 this chapter but adjusted to the RNA design problem. Our im-
2529 plementation, called aRNAque is available on GitHub as a python
2530 script. Compared to existing inverse folding tools, the benchmark
2531 results show improved performance on both pseudoknot-free
2532 and pseudoknotted datasets.

2533 5.1 MATERIAL AND METHODS

2534 This section provides a detailed description of aRNAque algorithm
2535 in general and in particular the Lévy mutation scheme imple-
2536 mented.

2537 5.1.1 *aRNAque's mutation operator*

2538 For a given target RNA secondary structure in its string represen-
2539 tation σ^* of length L , the space of potential solutions to the inverse
2540 folding problem is $\{A, C, G, U\}^L$. An evolutionary algorithm ex-
2541 plores the space of solutions through its move (or mutation)
2542 operator. To explore the search space of compatible sequences
2543 (sequences with canonical base pairs at the corresponding open
2544 and closed bracket positions) with σ^* exclusively, we propose a

2545 mutation step that depends on the nucleotide canonical base pair
 2546 probability distribution.

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

$$P_{\mathcal{N}} = \{w_A, w_C, w_U, w_G\}$$

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

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

where

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

2547 Our evolutionary algorithm relies on the flexibility of the mutation
 2548 parameters $P_{\mathcal{N}}, P_{\mathcal{C}}$. These parameters allow explicit control
 2549 of the GC-content of the RNA sequences during the designing
 2550 procedure.

2551 We examined the binomial and Zipf distributions:

- Binomial mutation: here U has a binomial distribution:

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

2552 for some $0 \leq \mu \leq 1$, such that $u = \mu \cdot L$. We can think of this
 2553 mutation mode arising from each nucleotide of an RNA
 2554 sequence independently undergoing a point mutation with
 2555 probability μ , i.e. μ is the per-nucleotide or point mutation
 2556 rate.

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

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

2557 where $c > 0$ is the value of the exponent characterizing the
 2558 distribution.

2559 Figure 5.1 shows the distribution of the number of point mutations
 2560 on a sequence of length 88 nucleotides for both mutation
 2561 schemes. Both distributions have the same mean, and the differ-
 2562 ence between the two distributions is more perceptible on their
 2563 tails.

2564 In the rest of this work, a local mutation will refer to a binomial
 2565 mutation with parameter $\mu \approx 1/L$.

2566 We present the mutation algorithm in Algorithm 1.

Algorithm 1: aRNAque's mutation algorithm

```

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

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

Output: P'

- 1 $\{B_i\} \sim \mathcal{D}(p, L)$, 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
- 2 $\{U_i\} \sim \mathcal{D}(p, L)$, 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
- 3 **for** $i \in \{1, 2, \dots, n\}$ **do**
 - 4 $\phi' \leftarrow P_i$; // Assign the sequence $\phi_i \in P$ to ϕ'
 - 5 **for** $j \in \{1, 2, \dots, U_i\}$ **do**
 - 6 $r \in \{1, 2, \dots, L\} \sim \mathcal{U}$; // select uniformly a random position in the RNA sequence ϕ'
 - 7 $n_j \in \{A, U, C, G\} \sim P_N$; // select a random nucleotide n_j with respect to P_N
 - 8 $\phi'_r \leftarrow n_j$; // replace the nucleotide at position j in the RNA sequence ϕ' with n_j
 - 9 **for** $j \in \{1, 2, \dots, B_i\}$ **do**
 - 10 $k_j \in \{AU, UA, CG, GC, UG, GU\} \sim P_C$; // select a random base pair k_j with respect to P_C
 - 11 $b \in \{(b_1, b_2)_i\} \sim \mathcal{U}$; // select uniformly a random pair of base pair positions
 - 12 $\phi'_b \leftarrow k_j$; // replace respectively the nucleotides at the base pair position $b_i \in b$ by $k \in k_j$
 - 13 $P' \leftarrow P' \cup \phi'$; // Add ϕ' to the list P'

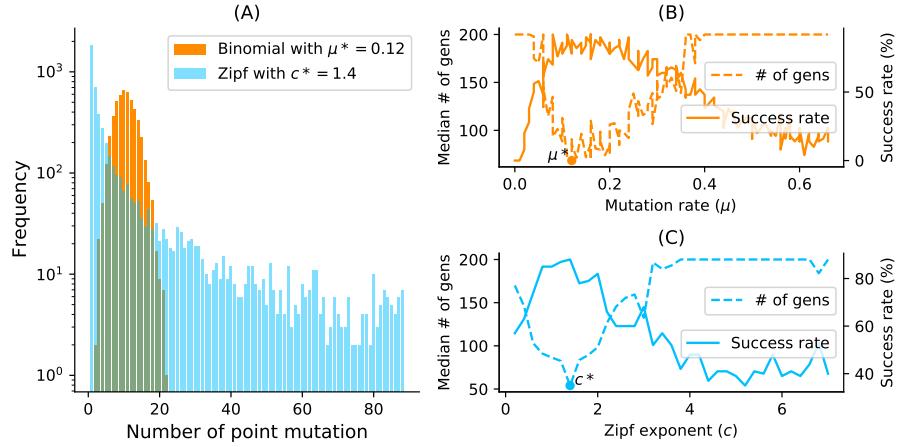


Figure 5.1: Binomial vs. Zipf distributions. (A) Samplings Binomial and Zipf distributions for the best binomial mutation rate μ^* (respectively c^* for the best Zipf exponent parameter). Both distributions have a mean of 8.7 point mutations for a sequence of length 88 nucleotides. (B) Tuning of binomial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

2567 5.1.2 aRNAque's objection functions

2568 Our EA reaches its performance through the minimization of
2569 three objective functions:

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

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

2570 where $d_h(\cdot, \cdot)$ is the hamming distance on the structure
 2571 space (structures are in dot and bracket representation)
 2572 defined in Equation 1.12.

- 2573 • Normalized Energy Distance (NED): It is used to minimize
 2574 the free energy of the designed sequences. (See Equation
 2575 1.14)
 - 2576 • Ensemble defect (ED) [194]: Here, we use the ED as a sec-
 2577 ond objective function for refinement after having at least
 2578 one sequence that folds into the target in the current popu-
 2579 lation. It is defined in Equation 1.13.
- 2580 To minimize the NED and the hamming distance of a population
 2581 of RNA sequences, instead of combining both measurements
 2582 to form a multi-objective function, we use them separately at a
 2583 different level of our EA. We use the NED as a selection weight for
 2584 the sequences that will be mutated, and the hamming distance is
 2585 used as a weight to elite ten best sequences that will always move
 2586 to the next generation. Therefore the selection method we use
 2587 is the *fitness proportionate selection*, also known as roulette wheel
 2588 selection [95]. Once we have at least one sequence that folds into
 2589 the given target in the current population (for the successful case),
 2590 we start a random walk in its neutral network by minimizing the
 2591 ensemble defect function (Eq. (1.13)). The next section provides
 2592 more detailed information about the core of our EA and the full
 2593 pseudo-code.

2594 5.1.3 *aRNAque's EA*

2595 For a given population size n and a target structure \mathcal{S}^* of length
 2596 L , an initial population P is generated randomly as follows:

- 2597 1. Select randomly L nucleotides in \mathcal{N}
- 2598 2. Identify the base pair position (i, j) in the random sequence,
 2599 select randomly a base pair in the set of canonical base pairs
 2600 \mathcal{C} and fix the first nucleotide of the selected canonical base
 2601 pair at the position i and the second at position j .
- 2602 3. Repeat 2. for all base pair positions in the target structure
- 2603 4. Repeat 1. 2. and 3. n -times.

2604 Let T be the maximum number of generations and F_t the set of
 2605 all sequences found at a given time t . After the initial population
 2606 of RNA sequences is generated, our algorithm is described in
 2607 Algorithm 3.1. The stopping criteria are two: 1) the number of
 2608 generations (t) is equal to the max number of generations (T)
 2609 or 2) the minimum hamming (or base pair) distance of the best
 2610 RNA sequence solution to the target is 0 (i.e the maximum fitness
 2611 value is 1).

2612 **5.1.4 Benchmark parameters and protocols**

2613 For the benchmark results presented in this work, we use three
 2614 datasets: the Eterna100 dataset, RFAM dataset and PseudoBase++
 2615 dataset. Depending on the datasets, a specific RNA folding tool
 2616 is used. This section gives more details about aRNAque's parame-
 2617 ters, energy parameters and other tools parameters used for the
 2618 benchmark results presented in this chapter.

2619 ***Folding tools***

2620 Two tools for pseudoknotted RNA folding are considered in this
 2621 work: HotKnots and IPknot. For pseudoknot-free RNA folding,
 2622 we used RNAfold. For the mutation parameter and GC-content
 2623 analysis presented in our work, we used IPknot, and both HotKnots
 2624 and IPknot for PseudobBse++ benchmarks. To be able to use
 2625 HotKnots in aRNAque without copying aRNAque in the bin direc-
 2626 tory of Hotknots, we have performed some modifications on
 2627 Hotknots source code. Details on the modifications are provided
 2628 in the supplementary material SI 6. Furthermore, we considered
 2629 pkiss, a well known tool for K-type pseudoknot prediction, but
 2630 since the PseudoBase++ dataset contains just 4 K-type pseudoknot-
 2631 ted structures and pKiss has higher time complexity ($O(n^6)$), we
 2632 did not find it efficient for the benchmark we performed.

2633 ***Mutation parameters tuning***

2634 The main challenge for an evolutionary algorithm is to find opti-
 2635 mum parameters such as mutation rate, population size and se-
 2636 lection function. We used 80 pseudoknotted targets with lengths
 2637 from 25 to 181 nucleotides for the mutation parameter analysis.
 2638 We set the maximum number of generations T to 200 and the
 2639 population size n to 100. The stopping criteria are two: 1) the
 2640 number of generations (t) is equal to the max number of gener-

Algorithm 2: aRNAque

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

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

Output: Best population P'

```

1  $P' \leftarrow P$ ; // Assign the initial population to the best
    population
2  $t \leftarrow 0$ ; // Initialize the number of generations to 0
3 while  $t \leq T$  &  $f(\sigma^{MFE}(\phi_b), \sigma^*) \neq 1$  do
4    $\Sigma \leftarrow \{\arg \min_{\sigma \in \Sigma} \Delta G(\phi_i, \sigma, \mathcal{P})\};$  // Fold each sequence
       $\phi_i \in P'$  and store them in  $\Sigma$ . Where  $i \in \{1, 2, \dots, n\}$ ,  $\Gamma$ 
      the structural ensemble and  $\Delta G(\phi_i, \sigma)$  the free energy
      computed using the parameters  $\mathcal{P}$ 
5    $\kappa = \lfloor (n \times 0.1) \rfloor$ ; // The number of RNA sequences to copy
      in the next generation without mutating them.
6    $F \leftarrow \{f(\sigma, \sigma^*) | \forall \sigma \in \Sigma\}$ ; // Evaluate the fitnesses of
      the folded population to the target strucre  $\sigma^*$  and
      store them in a list  $F$ 
7    $E_\kappa \leftarrow \{\phi_1 \dots \phi_\kappa\} \sim F$ ; // copy of the 10% best sequence
      based on their fitnesses  $F$ .
8    $P_S \leftarrow \{\phi_i\} \sim F$ , where  $i \in \{1, 2, \dots, n - \kappa\}$ ; // Randomly
      sample  $(n - \kappa)$  RNA sequences from  $P'$  with respect to
      their fitnesses  $F$ .
9    $M \leftarrow \text{mutate}(P_S, \mathcal{D}(p, L), P_C, P_N)$ ; // Mutated the
      selected sequences using the mutation algorithm
      presented in the main text in out paper.
10   $P_b \leftarrow M \cup E_\kappa$ ; // Combine the mutated population and the
      best solutions to form the new population that will be
      evolved in the next generation
11   $\phi_b \leftarrow \arg \max_{\sigma \in \Sigma} f(\sigma, \sigma^*)$ ;
12   $t \leftarrow t + 1$ ; // Increment the time step (the number of
      generations)

```

2641 ations (T) or 2) the minimum hamming (or base pair) distance
 2642 of the best RNA sequence solution to the target is 0. The best
 2643 mutation parameters (c^* for Levy and μ^* for Binomial) are those
 2644 that have the lowest median number of generations. The best mu-
 2645 tation parameters obtained for both binomial and Lévy mutation
 2646 modes are used to benchmark and compare the results on the
 2647 entire datasets of RNA structures.

2648 *Benchmark on the PseudoBase++ dataset*

2649 Four benchmarks are performed on the pseudoknotted dataset:
 2650 1) mutation parameter analysis, 2) the GC-content and diversity
 2651 analysis, 3) Local search versus Lévy search, 4) aRNAque (Lévy
 2652 search) versus antaRNA. For the aRNAque (Binomial and Lévy)
 2653 case, the four benchmarks share the same number maximum
 2654 number of generations ($T = 200$), population size ($n = 100$),
 2655 stopping criteria ($t = T$ or min fitness equals 0). For the antaRNA
 2656 benchmark, the maximum number of iterations was set to 1200,
 2657 and a slight modification was made to allow the support of the
 2658 folding tool HotKnots (See Appendix 6). For booth tools and each
 2659 benchmark, 20 runs were launched independently in parallel on
 2660 a computer with the same resources, resulting in 20 designed
 2661 sequences per pseudoknotted target structure. To measure the
 2662 performance of each tool, each designed sequence s is folded into
 2663 a secondary structure \mathcal{S} and the similarities between \mathcal{S} and \mathcal{S}^* are
 2664 computed using the base pair distance. For the GC-content bench-
 2665 mark, four GC-content values are considered, $\{0.25, 0.5, 0.75, 1\}$
 2666 and the setting of each tool remains the same.

2667 *Benchmark on the Eterna100 dataset*

2668 We performed two benchmarks are one the Eterna100 dataset: 1)
 2669 a benchmark on the Eterna100-V1 dataset using the Turner1999
 2670 energy parameter and the both versions of aRNAque (one point
 2671 and Lévy mutation), 2)a benchmark on the Eterna100-V2 dataset
 2672 using the Turner2004 energy parameter and both versions of
 2673 aRNAque (one point and Lévy mutation). For each of the Eterna100
 2674 benchmark we used the same evolutionary algorithm parameters;
 2675 a maximum of $T = 5000$ generations (i.e. a maximum of 500,000
 2676 evaluations), a population size of $n = 100$ and the same stopping
 2677 criteria (the number of generation $t = T$ or min fitness equals
 2678 0). For both local and Lévy search, 5 runs were launched inde-
 2679 pendently, which results in 5 designed sequences per target. We

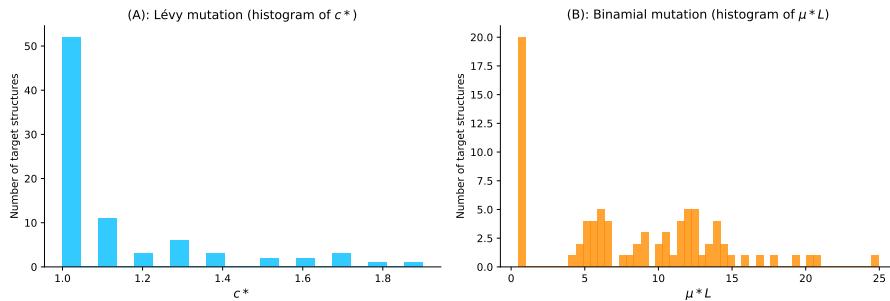


Figure 5.2: Parameter tuning for both binomial and Lévy mutation schemes. (A) Lévy mutation parameter tuning. Histogram of best exponent parameter (c^*) for a set of 81 target structures with different pseudoknot patterns and various lengths. The most frequent best exponent value is 1. (B) Binomial parameter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

2680 define success rate simply as the number of successfully designed
 2681 targets. A target is considered successfully designed when at least
 2682 one of the designed sequences folds into the target structure.

2683 For the benchmarks performed on ERD, NUPACK, and SentRNA the
 2684 default parameters were used. For NEMO, the number of iteration
 2685 was set to 2500 and for RNAinverse the objective function was set
 2686 to be the partition function and the number of iteration at 1200.

2687 *Benchmark on the non-Eterna100 dataset*

2688 For the non-Eterna100 dataset, only the Turner2004 energy param-
 2689 eters were used. The maximum number of generations was set
 2690 to be 5000. The mutation parameters (P_C and P_N) were chosen
 2691 to be close to the nucleotide distribution of the RNA sequence in
 2692 nature [43].

2693 5.2 EXPERIMENTAL RESULTS

2694 5.2.1 aRNAque's performance on pseudoknot-free target structures

2695 We compared the performance of aRNAque for pseudoknot-free
 2696 target using the benchmark datasets: the non-Eterna100 and
 2697 the Eterna100. This subsection presents the statistical results

2698 obtained compared to benchmarked existing tools and the results
 2699 found in the literature. In addition, we compared the performance
 2700 of aRNAque (Lévy mutation) to the one of Ivry et al. [77] on a
 2701 tripod pseudoknot-free RNA secondary structure.

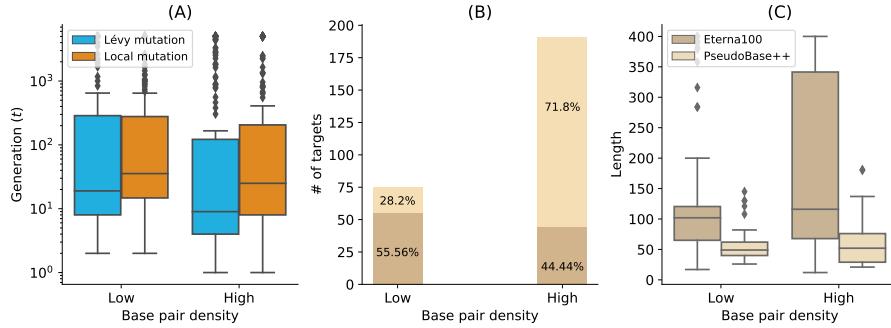


Figure 5.3: **Lévy mutation vs. Local mutation: performance analysis with respect to the base-pair density.** The higher the base-pair density is, the more useful the Lévy mutation scheme to speed up the optimization EA. (A) Distributions of number of generations for the low and high base-pair density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

2702 5.2.1.1 Performance on Eterna100 dataset

2703 A first benchmark was performed on the Eterna100 datasets. First,
 2704 on the Eterna100-V1 dataset, the Lévy flight version of aRNAque
 2705 successfully designed 89% of the targets and the one-point muta-
 2706 tion (local mutation) version achieved 91% of success, suggesting
 2707 that for some target structures, local mutation can outperform the
 2708 Lévy mutation scheme. Combining the two solutions, aRNAque
 2709 solved in total 92% of the targets of Eterna100-V1.

2710 When analysing the performance of Lévy flight for low and
 2711 high base pair densities separately, the median number of gener-
 2712 ations of high base pair density targets was lower than the one
 2713 with low base-pair density (8 generations for high density and
 2714 18 for the low base pairs density targets). The same observation
 2715 was drawn for the success rate. For the low base-pair density
 2716 targets, the Lévy flight achieved 87% (49/56) success whereas,
 2717 for the high base-pair density, it achieved 91% (40/44). The same
 2718 analysis can be done when comparing the one-point mutation

2719 results for the high-density targets to the Lévy flight mutation.
2720 The median number of generations for the low-density targets
2721 when using a one-point mutation operator was 34 (respectively
2722 24 for the high base pair density targets) (see Figure 5.3A).

2723 Another benchmark was performed on Eterna100-V2 with
2724 aRNAque achieving a 93% success rate when combining the de-
2725 signed solutions for both mutation schemes. Compared to re-
2726 cently reported benchmark results [92], aRNAque achieved almost
2727 similar performance to NEMO on Eterna-V2: one target was un-
2728 solved by all existing tools and one target solved only by NEMO
2729 remained unsolved by aRNAque, outperforming all existing EA
2730 methods.

2731 For the robustness analysis, Table 5.1 presents the benchmark
2732 results on Eterna100-V1 using the Turner2004 energy parame-
2733 ters sets. It shows that the evolution algorithm we propose can
2734 solve $\approx 72\%$ of the dataset, and it surpasses the 4 methods we
2735 benchmarked and all the tools already benchmarked in [151].
2736 We can also solve approximately 23 targets more than NUPACK,
2737 which is also minimizing the ensemble defect and that shows the
2738 importance of a population-based algorithm. Compared to the
2739 existing GA-based algorithms, our EA can solve approximately
2740 18 targets more than MODENA and 7 targets more than ERD.

2741

2742 5.2.1.2 Performance on non-Eterna100

2743 Additionally to the Eterna100 dataset, we also used the non-
2744 EteRNA dataset collected from the RFAM database to assess the
2745 aRNAque's performance on pseudoknot-free target secondary struc-
2746 ture. Compared to other tools, the statistical results are presented
2747 in Table 5.2.

2748 The results show that our method surpasses 8/10 of other
2749 tools. ERD solved 2 more targets than our method because of its
2750 strong decomposition capacity, which allows it to solve the entire
2751 **dataset B**. With the advantage that our evolutionary algorithm
2752 also allows us to fit the nucleotide distribution parameters taken
2753 from natural RNA directly in the mutation parameters, we can
2754 solve 21/24 targets from the **dataset B**. For the **dataset A** aRNAque
2755 solves 24/29 targets which means 2 more than the existing tools
2756 and for the 10 last targets, it solves 7 targets. Adding all these
2757 solved targets together, we obtain a result of 52/63 as presented
2758 in Table 5.2.

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

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

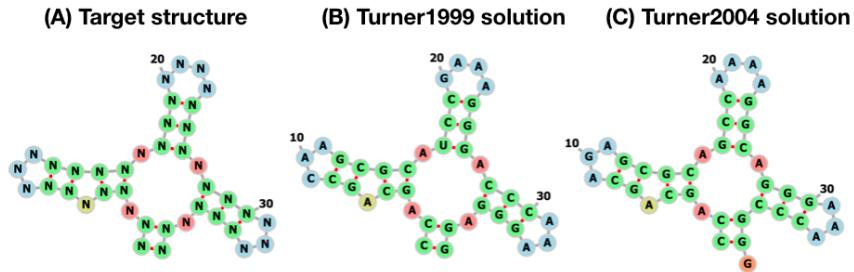


Figure 5.4: aRNAque’s performance on a TRIPOD secondary structure. (A) The tripod target structure. (B) aRNAque’s solution using the Turner1999 energy parameter sets. (C) aRNAque’s solution using the Turner2004 energy parameter sets.

2759

2760 5.2.1.3 aRNAque performance on a tripod secondary structure

2761 Finally, we performed a benchmark on a tripod target secondary
 2762 structure. The tripod secondary structure was used as a third test
 2763 case in the work of Ivry et al. [77], and it does not contain any
 2764 pseudoknot interactions. It comprises four stems, three of which
 2765 with terminal hairpins, surrounding a multibranch loop (See Fig-

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

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

ure 5.4A). The tripod target structure was proved to be very challenging, especially because of its multiloop component, which is also found in some of the unsolved Eterna100 target structures. We perform here, for both energy parameters Turner1999 and Turner2004, 100 independent designs, using a population size of 100 RNA sequences and a maximum of 5000 generations. The mutation parameters used are: $P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$, $P_N = \{0.7, 0.1, 0.1, 0.1\}$ and $c = 1.5$. When using the Turner2004 energy parameter set, none of the 100 designed RNA sequences was successful (i.e, 0 sequence folds exactly into the target structure after 5000 generations). Figure 5.4B shows one of the best solutions obtained out of 100 designed sequences when using the Turner2004, the designed sequence folds into a structure at one error base-pair distance from the target structure. In contrast, when using the Turner1999 energy parameters, we successfully designed the tripod secondary structure (See Figure 5.4C). The 100 sequences designed folded exactly into the target structure with an average median number of generations 20. When comparing both solutions to the one obtained in [77], aRNAque (with no need of changing the RNA structure distance) can successfully

2786 design the multibranch loop component with one base pair er-
 2787 ror using the Turner2004 energy parameter whereas RNAinverse
 2788 (with the DoPloCompare distance) failed to design the multi-
 2789 branch loop, and the solution was at 2 base-pair distance error.

2790 5.2.2 *aRNAque's performance on pseudoknotted target structures*

2791 Secondly, we assessed the performance of aRNAque in designing
 2792 pseudoknotted target secondary structures through intensive
 2793 benchmark on PseudoBase++ dataset. We then compared the re-
 2794 sults obtained to the one of antaRNA, using both folding tools
 2795 Hotknots and IPknot. Furthermore, a comparison between local
 2796 and Lévy mutations is provided.

2797 5.2.2.1 *Best mutation parameter analysis on PseudoBase++: Levy
 2798 mutation vs. local mutation*

2799 The advantage of using a Lévy mutation is its capacity to allow
 2800 simultaneous search at all scales over the landscape. The search
 2801 at different scale is often dictated by the exponent parameter of
 2802 the heavy-tailed distribution. In this first subsection, we analyse
 2803 for 80 pseudoknotted target structures and for both mutation
 2804 schemes the distributions of the best mutation parameters.

- 2805 • Binomial mutation: From Figure 5.1B, the critical range was
 2806 identified to be from 0 to 0.2 and as μ becomes greater
 2807 than 0.1, the success rate decreases and the average number
 2808 of generations increases. For each of the 80 target struc-
 2809 tures with pseudoknots, 20 sequences were designed for
 2810 $\mu \in [0, 0.2]$ with a step size of $1/L$. Figure 5.2B shows the
 2811 histogram of the best mutation rate found for each target
 2812 structure. Two main regimes are apparent: one where the
 2813 best mutation rate is very low mutation rate ($\approx 1/L$) and
 2814 another where the high mutation rate is optimal.
- 2815 • Lévy mutation: From Figure 5.1C, the critical range of c was
 2816 identified to be $[1, 2]$. For $c \in [1, 2]$ and a step size of 0.1,
 2817 an optimum exponent parameter c^* was investigated for all
 2818 the 80 target structures. Figure 5.2A shows the histogram of
 2819 c^* . Contrary to binomial mutation, the optimum exponent
 2820 parameter does not vary too much ($\forall \mathcal{S}, c^* \approx 1$).

2821 Figure 5.2 shows that when using a Lévy mutation, the optimal
 2822 mutation rate is the same for most target structures. In contrast,

2823 the optimum binomial mutation rate parameter μ^* mostly varies
 2824 with different targets. Although both mutation schemes (for the
 2825 best mutation parameters) have approximately the same success
 2826 rates, the Lévy flight mutation scheme is more robust to different
 2827 targets.

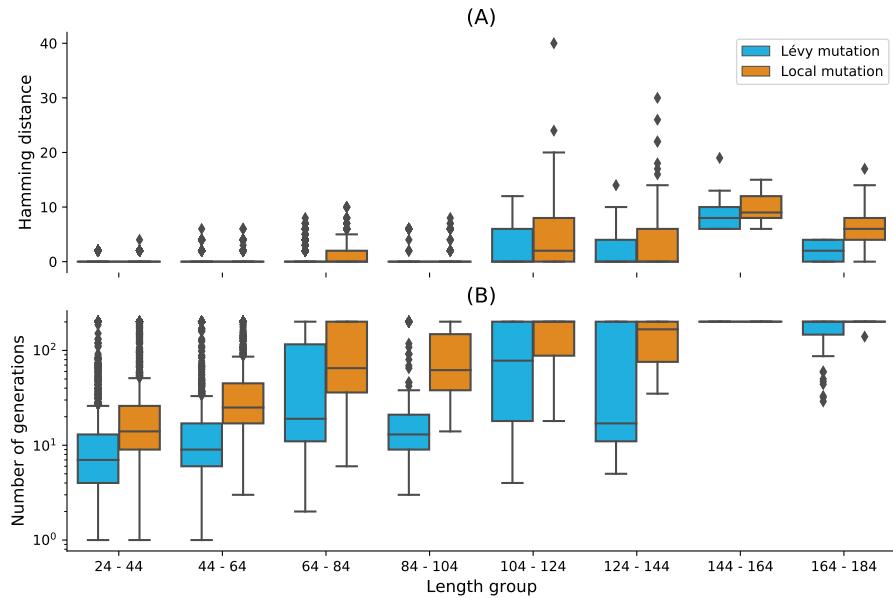


Figure 5.5: **Lévy mutation mode vs local mutation (one-point mutation).** (A) Hamming distance distributions *vs.* target structure lengths. (B) Number of generations distributions for different length groups. In both (A) and (B), lower values indicate better performance. The target structures are solvable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

2828 **5.2.2.2 Performance on PseudoBase++: Levy mutation vs. local mu-
 2829 tation**

2830 Figure 5.5 shows box plots for the base pair distance (Hamming
 2831 distance) and the number of generations for increasing target

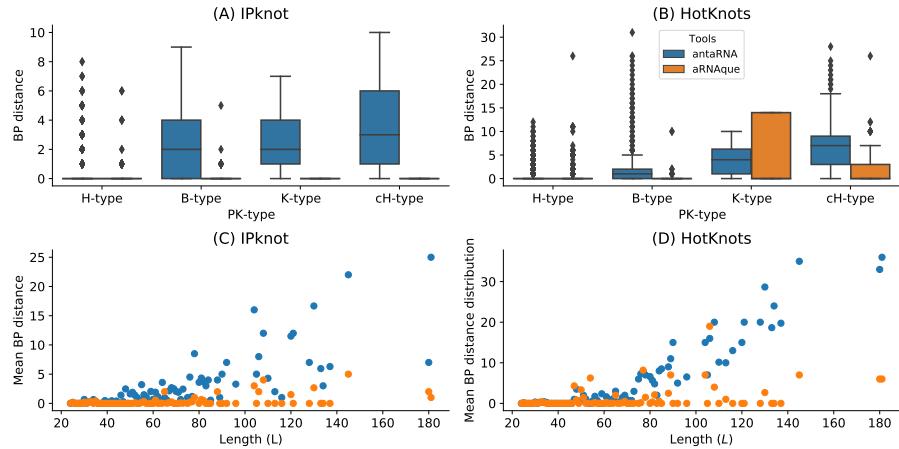


Figure 5.6: aRNAque vs antaRNA on PseudoBase++ dataset using both IPknot and HotKnots. Lower values imply better performance. (A, B) Base pair distance distributions of the designed sequences to the target structure for different pseudoknot types. (C,D) Mean base pair distance against target lengths.

lengths under our two mutation schemes: binomial at low mutation rate (or one point mutation) and the Lévy mutation. For each pseudoknotted RNA target structure in the PseudoBase++ dataset, we designed 20 sequences. The results show that using the Lévy mutation instead of a local mutation scheme can significantly increase the performance of aRNAque. The gain was less significant in terms of designed sequences quality (base pair distance distributions, with a t -value ≈ -1.04 and p -value ≈ 0.16) but more significant in terms of the average number of generations needed for successful matches to target structures (with a t -value ≈ -3.6 and p -value ≈ 0.0004). This result demonstrates a substantial gain in computational time when using a Lévy mutation scheme instead of a purely local mutation.

5.2.2.3 Performance on PseudoBase++: aRNAque vs. antaRNA

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

2854 dian base-pair distance distribution increases). On the other hand,
 2855 aRNAque's performance improves as pseudoknot complexity in-
 2856 creases (e.g. the mean base-distance decreases with the pseudo-
 2857 knot complexity).

2858 A second benchmark using HotKnots as a folding tool was
 2859 performed on the same dataset. For both aRNAque and antaRNA,
 2860 the more complex the pseudoknot motifs, the worse is the tool
 2861 performance (median of the base-pair distance distribution in-
 2862 creases). Figures 5.6B and 5.6D show the base pair distance distri-
 2863 butions with respect to the pseudoknot motifs for both aRNAque
 2864 and antaRNA. Even though both performances degrade as tar-
 2865 get length increases, aRNAque (Lévy flight evolutionary search)
 2866 performance remains almost constant for all the target lengths
 2867 greater than 60.

2868 5.2.3 *Quality of the designed RNA sequences*

2869 In addition to the successful rate analysis, we assessed the quality
 2870 of the designed RNA sequences by analysing both GC-content
 2871 and diversity of the pseudoknotted dataset using IPknot. This
 2872 section presents the results obtained and a comparison to antaRNA
 2873 designed sequences.

2874 5.2.3.1 *GC–content analysis of the designed sequences using IPknot*

2875 The GC–content of an RNA sequence S measures the concen-
 2876 tration of G-C nucleotide in S and influences its stability and
 2877 biological function. Therefore, the ability of an inverse folding
 2878 tool to control the GC–content is of vital importance for designing
 2879 functional RNA sequences. Both antaRNA and aRNAque allow to
 2880 control the GC–content at different levels of the optimization
 2881 process: aRNAque through the mutation parameters P_C and P_N ;
 2882 antaRNA with the parameter $tGC \in [0, 1]$. In this section, we com-
 2883 pare the performance of each tool for fixed GC–content values
 2884 and analyse each tool's ability to control the GC–content. For
 2885 each pseudoknotted target structure in the PseudoBase++ dataset,
 2886 four different GC-content values $\{0.25, 0.5, 0.75, 1\}$, a poll of 20
 2887 sequences is designed using IPknot as folding tool. That results
 2888 in 5320 designed sequences for each GC-content value and tool.
 2889 The number of successes is the total number of sequences that
 2890 fold exactly into the given target structure (i.e. the designed se-
 2891 quence folds into a structure at base-pair distance 0 from the
 2892 target structure). Figure 5.7 shows respectively the base pair dis-

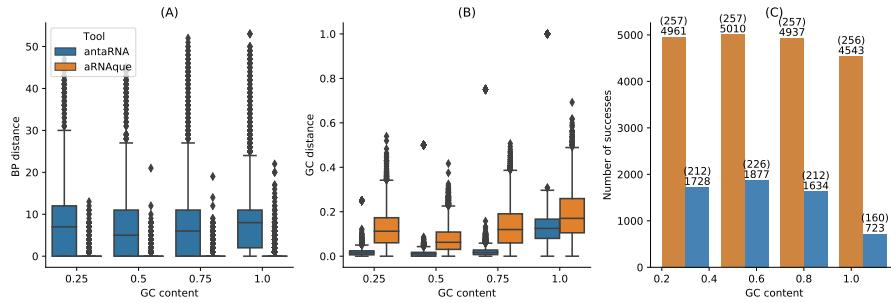


Figure 5.7: **aRNAque vs antaRNA on PseudoBase++ dataset using IPknot: GC-content analysis.** (A) Base-pair distance distributions. (B) GC-content distance distributions. The difference between the targeted GC-content and the actual GC-content values. In (A,B), lower values imply better performance.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

tance distributions, the GC distance distributions and the number of successes for both aRNAque and antaRNA. The results show that the performance (in terms of success number) varies considerably with the GC-content values for both tools, and the best performance is obtained for both tools with a GC-content value of 0.5. When comparing the GC-content distance (i.e absolute value of the difference between the targeted GC-content and the actual GC-content values of the designed sequences) distributions, both GC-content distance median distributions increase, whereas antaRNA controls significantly better the GC-content (See Figure 5.7B). On average, for the respective GC-content values {0.25, 0.5, 0.75, 1}, antaRNA's sequences have respectively 0.2569, 0.4952, 0.7314, 0.8684 whereas aRNAque's sequences have respectively 0.3649, 0.4910, 0.6231, 0.811; the main difference is at fixed GC-content values 0.25 and 0.75. Even though antaRNA designs sequences with better control of the GC-content, the gap in success rate still remains remarkable compared to aRNAque (See Figure 5.7A and Figure 5.7C).

5.2.3.2 Diversity of the designed sequences

Another advantage of using a Levy mutation when designing RNA sequences is to increase the chance of designing sequences

with high diversity. Here, we use the positional entropy of each pool of 20 sequences previously designed for each pseudoknotted target structure to compare the diversity of RNA of both tools antaRNA and aRNAque (Lévy search). We also compare it to the diversity of the designed sequences using the old version of aRNAque (Local search). The results show that the sequence diversity of both antaRNA and aRNAque (Lévy search) varies with the GC-content values, where the more diversified pool of sequence is achieved with a GC-content value of 0.5. When comparing the pool of designed sequences with highest entropy (i.e. with a fixed GC-content of 0.5) to the one of the old version of aRNAque (Local search), the aRNAque (Lévy search) and antaRNA produce sequences with similar entropy (i.e. with a median entropy of 61.01 for Lévy search respectively 59.65 for antaRNA (see Figure 5.8), whereas the entropy of the sequences designed using the Local search is lower. For the three others fixed GC-content values (i.e. {0.25, 0.75, 1}, aRNAque (Lévy search) produces sequences with the highest entropy (respectively a median entropy of 58.9, 60.08, 51.52 against 53.42, 54.63, 48.38 for antaRNA).

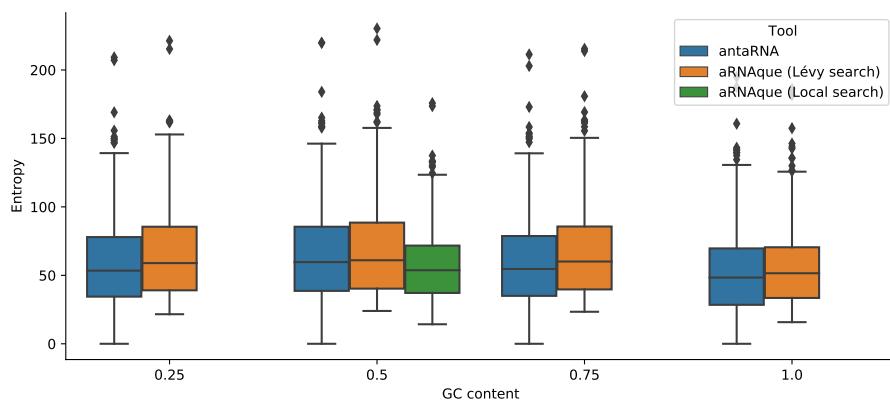


Figure 5.8: aRNAque vs antaRNA on PseudoBase++ dataset using IPknot: Diversity analysis. The positional entropy distributions plotted agains the targeted GC-content values. Higher values imply better performance.

2932

2933 5.2.4 Complexity and CPU time comparison

2934 We finally analysed the design performance of aRNAque relatively
 2935 to the CPU time needed. This section presents aRNAque statistical
 2936 results compared to two main tools: RNAinverse for the
 2937 pseudoknot-free targets and antaRNA for the pseudoknotted tar-
 2938 gets.

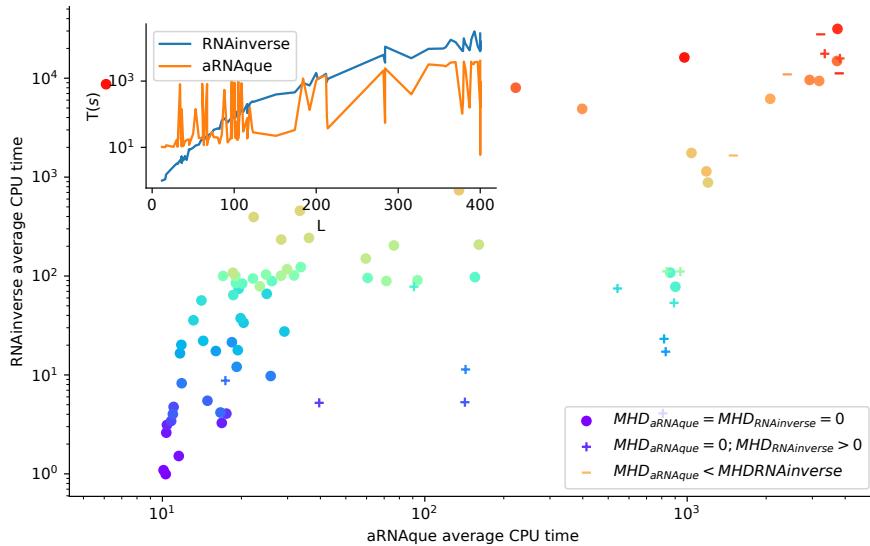


Figure 5.9: **CPU time: RNAinverse vs. aRNAque.** Each bubble corresponds to a target structure in EteRNA100 dataset and, their colours are proportional to the length of the targets. In the legend, MHD stands for Median Hamming distance, and the different markers represent—('o') 100% success for both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

2939 5.2.4.1 *CPU time vs. success rate using RNAfold: RNAinverse vs.
2940 aRNAque on EteRNA100-V1.*

2941 Since our previous benchmarks on EteRNA100-V1 using the Turner2004
2942 energy's parameters reveal that RNAinverse, one of the oldest in-
2943 verse folding tools, stands behind aRNAque solving 66% of the
2944 dataset; we have chosen to compare its computational time to our
2945 implementation (See Table 5.1). The inset of the Figure 5.9 shows
2946 the CPU time in seconds needed to design for each target in the
2947 EteRNA100-V1, 5 sequences. As the RNAinverse time increases
2948 exponentially with the length of the target, the aRNAque one does
2949 not.

2950 When comparing the ratio between the success rate and CPU
2951 time, aRNAque mostly succeeded in finding at least one sequence
2952 that folds to the target with lower CPU time costs for average
2953 target lengths. In contrast, RNAinverse accuracy is lower, and the

2954 CPU time is expensive. The increase in CPU time may be because
 2955 of the use of the partition function as the objective function.

2956 **5.2.4.2 CPU time vs. success rate using Hotknots: antaRNA vs.
 2957 aRNAque on PseudoBase++**

2958 We also compare aRNAque's computational time to the one of
 2959 antaRNA. For both tools, 20 sequences were designed for each
 2960 target structure of the PseudoBase++ dataset. The GC-content
 2961 value used for both tools is 0.5, and the maximum number of
 2962 interactions for antaRNA is 5000. Figure 5.10 shows the median
 2963 CPU time of the 20 runs in seconds for both tools plotted against
 2964 each other. We analysed the CPU time by partitioning the data
 2965 into three groups: 1) a set for which both tools have a median
 2966 base-pair distance of 0 (158 entries marked with 0); 2) another
 2967 set for which aRNAque has a median base-pair distance is 0 and
 2968 antaRNA (41 entries marked with +); 3) the last set for which
 2969 antaRNA designs are of better quality (9 entries mark as -). For
 2970 the first group, we can notice that for most targets of short length
 2971 antaRNA is faster than aRNAque. For the second group, although
 2972 antaRNA average CPU time remains smaller, aRNAque's success
 2973 rate outperformed antaRNA. On the one hand, aRNAque average
 2974 CPU time is higher than the one of antaRNA, but this could be
 2975 due to its population-based algorithm, which often allows for de-
 2976 signing more successful sequences. On the other hand, antaRNA
 2977 is faster but less successful. Increasing antaRNA's number of itera-
 2978 tions will indeed increase the CPU time, but it may improve the
 2979 quality of the designed sequences.

2980 **5.3 CONCLUSION**

2981 In this work, we investigated an evolutionary approach to im-
 2982 prove the existing solutions to the RNA inverse folding problem.
 2983 As a result, we proposed a new EA python tool called aRNAque.
 2984 aRNAque implements a Lévy flight mutation scheme and supports
 2985 pseudoknotted RNA secondary structures. The benefit of a Lévy
 2986 flight over a purely local (binomial with $\mu \ll 1$ or a single point
 2987 mutation) mutation search allowed us to explore RNA sequence
 2988 space at all scales. Such a heavy-tailed distribution in the num-
 2989 ber of point mutations permitted the design of more diversified
 2990 sequences.

2991 Our results show general and significant improvements in the
 2992 design of RNA secondary structures compared to the standard

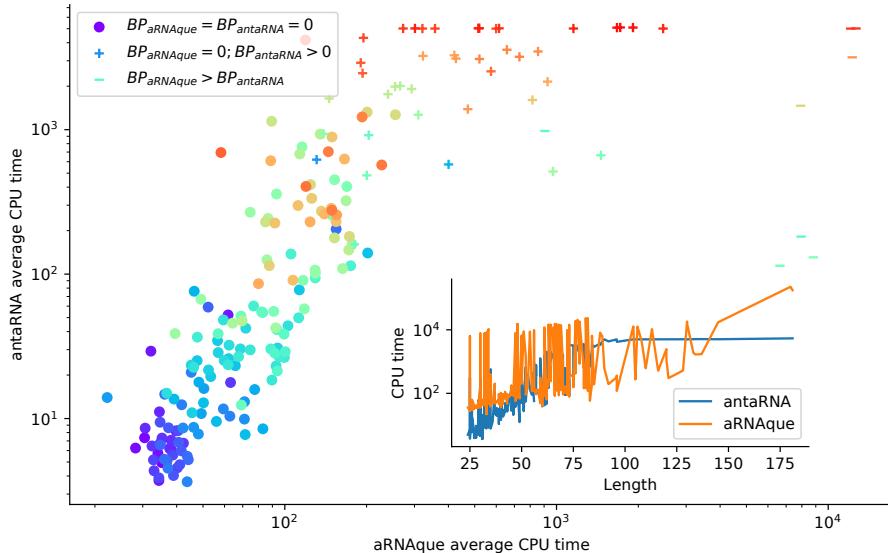


Figure 5.10: CPU time analysis using Hotknots: antaRNA vs. aRNAque.

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

2993 evolutionary algorithm mutation scheme with a mutation parameter $\approx 1/L$, where L is the sequence solution length. Lévy flight
 2994 mutations lead to a greater diversity of RNA sequence solutions
 2995 and reduce the evolutionary algorithm’s number of evaluations,
 2996 thus improving computing time.
 2997

2998

Part III

2999

GENERAL CONCLUSION AND DISCUSSIONS

3000

3001

[June 17, 2022 at 17:55 – 1.0]

3002

3003 LIMITATIONS OF THE PROPOSED METHODS
3004 AND PERSPECTIVES

3005 In the presented thesis, we have provided the molecular biol-
3006 ogy background and the biological functions of nucleic acids,
3007 especially non-coding RNAs. Because of the implication of the
3008 secondary structure of non-coding RNAs in performing biolog-
3009 ical functions, our study focuses on the secondary structure of
3010 ncRNAs. Therefore, we have introduced the concepts of RNA
3011 bioinformatics and the essential computational problems related
3012 to the secondary structure of ncRNAs, such as RNA folding and
3013 the inverse problem. We presented a wealthy literature review
3014 on existing tools that deal with both problems and some limi-
3015 tations for each tool. Despite very advanced results in the field,
3016 we have newly introduced two computation tools: RAFFT and
3017 aRNAque. What are the advantages and limitations of those tools?
3018 Is there any room for further improvements? How do these tools
3019 relate to evolutionary dynamics? In this concluding chapter of
3020 our thesis, we will try to provide an answer to these questions
3021 by first discussing the advantages and the limitations of the tools
3022 previously introduced.

3023 6.1 rafft: LIMITATIONS AND FUTURE WORKS

3024 In sum, RAFFT was first compared the algorithm performance for
3025 the folding task. Two structure estimates were compared with
3026 our method: the Thermodynamic-based tools computed using
3027 RNAfold, LinearFold, RNAstructure and the ML estimate using
3028 MxFold2 and CONTRAfold. When we considered the lowest energy
3029 structure, the comparison of RAFFT to existing tools confirmed
3030 the overall validity of our approach. In more detail, comparison
3031 with thermodynamic/ML models yielded the following results.
3032 First, the ML predictions performed consistently better than both
3033 RAFFT and other approaches, where the PPV = 70.4% and sensi-
3034 tivity = 77.1% on average. Second, the ML methods produced
3035 loops, such as long hairpins or external loops. We argue that
3036 the density of those loops correlate with the ones in the bench-
3037 mark dataset, which a PCA analysis revealed too. In contrast, the
3038 density of loops was lower in the structure spaces produced by

3039 RAFFT and other thermodynamic-based methods, implying some
 3040 over-fitting in the ML model. Finally, known structures obtained
 3041 through covariation analysis reflect structures *in vivo* conditions.
 3042 Therefore, the structures predicted by ML methods may not only
 3043 result from their sequences alone but also from their molecular
 3044 environment, e.g. chaperones. We expect the thermodynamic
 3045 methods to provide a more robust framework for the study of
 3046 sequence-to-structure relations. With respect to thermodynamic-
 3047 based tools, we obtained a substantial gain of performance when
 3048 analyzing $N = 50$ predicted structures per sequence and not only
 3049 the lowest energy one. This gain was even more remarkable for
 3050 sequences with fewer than 200 nucleotides, reaching the accuracy
 3051 of ML predictions.

3052 So how does RAFFT predictions contain structures that are more
 3053 relevant than the MFE, although these structures are less ther-
 3054 modynamically stable? The interplay of three effects may explain
 3055 this finding. First, the MFE structure may not be relevant because
 3056 active structures can be in kinetic traps. Second, RAFFT forms a set
 3057 of pathways that cover the free energy landscape until they reach
 3058 local minima, yielding multiple long-lived structures accessible
 3059 from the unfolded state. Third, the energy function is not perfect,
 3060 so that the MFE structures computed by minimizing it may not
 3061 in fact be the most stable.

3062 We also showed that the fast-folding graph produced by RAFFT
 3063 can be used to reproduce state-of-the-art kinetics, at least qual-
 3064 itatively. Our method demonstrated three main benefits. First,
 3065 the kinetics can be drawn from as few as 68 structures, whereas
 3066 the barrier tree may require millions. Second, the kinetics ansatz
 3067 describes the complete folding mechanism starting from the un-
 3068 folded state. Third, for the length range tested here, the proce-
 3069 dure did not require any additional coarse-graining into basins.
 3070 (Longer RNAs might require such a coarse-graining step, in
 3071 which structures connected in the fast-folding graph are merged
 3072 together).

3073 Based on our results, we believe that the proposed method is a
 3074 robust heuristic for structure prediction and folding dynamics.
 3075 The folding landscape depicted by RAFFT was designed to follow
 3076 the kinetic partitioning mechanism, where multiple folding path-
 3077 ways span the folding landscape. This approach has shown good
 3078 predictive potential. Furthermore, we derived a kinetic ansatz
 3079 from the fast-folding graph to model the slow part of the fold-
 3080 ing dynamics. It was shown to approximate the usual kinetics
 3081 framework qualitatively, although using many fewer structures.

3082 However, further improvements and extensions of the algo-
 3083 rithm may be investigated. First, the choice of stems is limited
 3084 to the largest in each positional lag, a greedy choice which may
 3085 not be optimal. Second, we have constructed parallel pathways
 3086 leading to diverse, accessible structures. Still, we have not given
 3087 any thermodynamic-based criterion to identify which are more
 3088 likely to resemble the native structure. We suggest using an ML-
 3089 optimized score to this effect. Our method can also find applica-
 3090 tions in RNA design, where the design procedure could start with
 3091 identifying long-lived intermediates and using them as target
 3092 structures. We also believe that mirror encoding can be helpful
 3093 in phylogenetic analysis. Indeed, the correlation spectra $\text{cor}(k)$
 3094 computed here contained global information of base-pairing that
 3095 can be used as a similarity measure. Finally, the versatile method
 3096 implemented in RAFFT gives possibilities for an alternative appli-
 3097 cation of the FFT in RNA-RNA interaction. The underlying idea
 3098 is that instead of encoding a sequence X and its mirror sequence
 3099 \bar{X} , one can consider two encoded sequences X and Y , and the
 3100 correlation between them will allow identifying the fraction of
 3101 high interaction between two RNA sequences quickly. In general,
 3102 RNA-RNA interaction prediction methods are divided into three
 3103 groups: alignment like methods, MFE methods and comparative
 3104 methods. MFE methods constitute the majority of the RNA-RNA
 3105 interaction tools, with the only difference often based on whether
 3106 the method considers intramolecular interactions. Some meth-
 3107 ods measure the accessibility of binding region (Intra and inter
 3108 interactions) [7, 32, 175]. We suggest neglecting intramolecular
 3109 interactions and intermolecular binding pairs for a preliminary
 3110 implementation.

3111 6.2 arnaque: LIMITATIONS AND PERSPECTIVES

3112 We have provided in the preview chapter, a new tool aRNAque,
 3113 implementing a Lévy flight mutation scheme that supports pseu-
 3114 doknotted RNA secondary structures. A Lévy mutation scheme
 3115 offers exploration at different scales (mostly local search com-
 3116 bined with rare big jumps). Such a scheme significantly improves
 3117 the number of evaluations needed to hit the target structure,
 3118 while better avoiding getting trapped in local optima. The bene-
 3119 fit of a Lévy flight over a purely local mutation search allowed
 3120 us to explore RNA sequence space at all scales. Such a heavy
 3121 tailed distribution in the number of point mutations permitted
 3122 the design of more diversified sequences and reduced the num-

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

We argue that the improved performance of the Lévy mutation over local search in target RNA structures is due to the high base pair density of pseudoknotted structures. Given that pseudoknots present a high density of interactions, there are dramatic increases in possible incorrect folds and thus increasing risk of becoming trapped near local optima [66]. 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 (den-

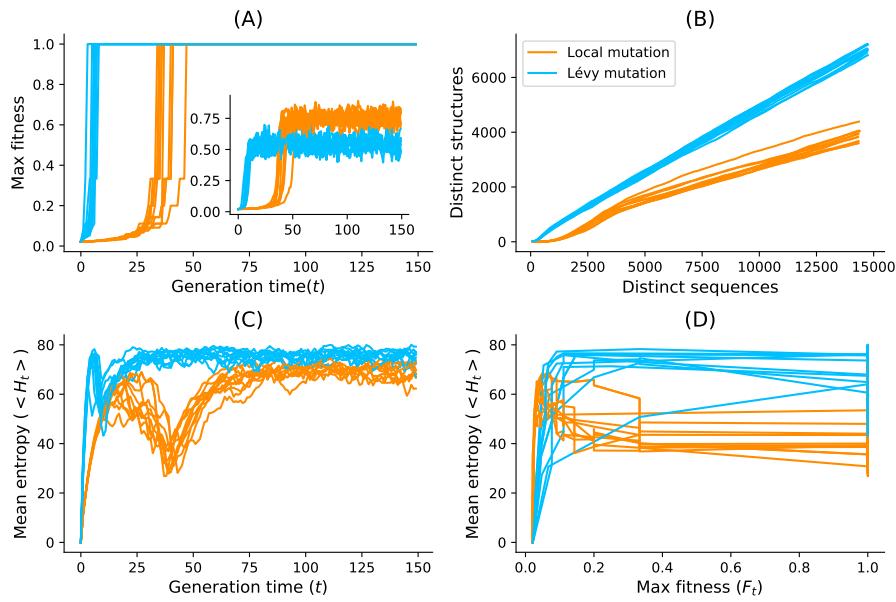


Figure 6.1: Lévy mutation vs one-point mutation. For the Eterna100 target structure [*CloudBeta*] 5 *Adjacent Stack Multi-Branch Loop*, ten independent runs were performed in which a minimum of 10 sequences were designed per run. (A) Max fitness and mean fitness (inset) over time. (B) Distinct sequences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

sity ≤ 0.5) and a second cluster for structures with high base pair density (density > 0.5). Figure 5.3B showed the number of target sequences available in each low and high density category. The number of targets available in each category are colored according to the percentage of pseudoknot-free targets (Eterna100-V1) vs. targets with pseudoknots (Pseudobase++), showing that pseudoknots are strongly associated with high base pair densities: 71% of the pseudoknotted target structures have a high base pair density. In contrast, the Eterna100 dataset without pseudoknots has somewhat higher representation at low base pair density. If it is true that improved Lévy Flight performance is indeed tied to base pair density, it is possible that similar heavy-tailed mutation schemes could offer a scalable solution to even more complex inverse folding problems. Another measure of difficulty is the length of the target RNA secondary structure. When analysing the mean length of the pseudoknot-free targets, the high base-pair density targets are on average 181 nucleotides longer, and the low-density base-pair targets are 139 nucleotides (See Figure

3184 5.3C). We have 49 nucleotides for low-density targets for the pseu-
 3185 doknotted targets and 52 nucleotides for the high-density targets.
 3186 That suggests that the Lévy mutation may be a good standard
 3187 for designing more challenging target structures.

3188 A further effort have been made to understand the cases in
 3189 which the Levy flight mutation can outperform the Binomial
 3190 with low mutation rate or a constant one-point mutation rate.
 3191 The key point of a Lévy mutation for the Inverse folding problem
 3192 partially may rely on the base-pair density and the stability of
 3193 stems with budge.

3194 To further illustrate that advantage, we considered the space
 3195 of all RNA sequences of length 12 and with only G,C nucleotides.
 3196 The structures with the lowest neutral set are:

- 3197 1. $T_1 = (((...)).))$: only 2 sequences fold into the secondary
 3198 structure T_1
- 3199 2. $T_2 = ((.((...))))$: only 1 sequence folds into the secondary
 3200 structure T_2

3201 When having a close look at those two structures the base pair
 3202 density is maximal and there is an unpaired position on both that
 3203 allows the formation of a budge.

3204 What that means naively is that any compatible sequence to T_1
 3205 (or T_2) will likely fold into a stem with four or three base pairs(
 3206 (((...))). Or (((....))..) , and these particular structures have
 3207 respectively 243 and 249 sequences in their neutral sets.

3208 We claim that, when having such kind of structure (T_1 or T_2),
 3209 the levy mutation is of an important role to get out of the huge
 3210 neutral network of more stable stems. A simple test case was to
 3211 run aRNAque for a target secondary structure T_1 . For both one
 3212 point and Lévy mutations, the distribution of the number of
 3213 generations needed to find sequences that fold into T_1 for both
 3214 mutation schemes is plotted in Figure 6.2.

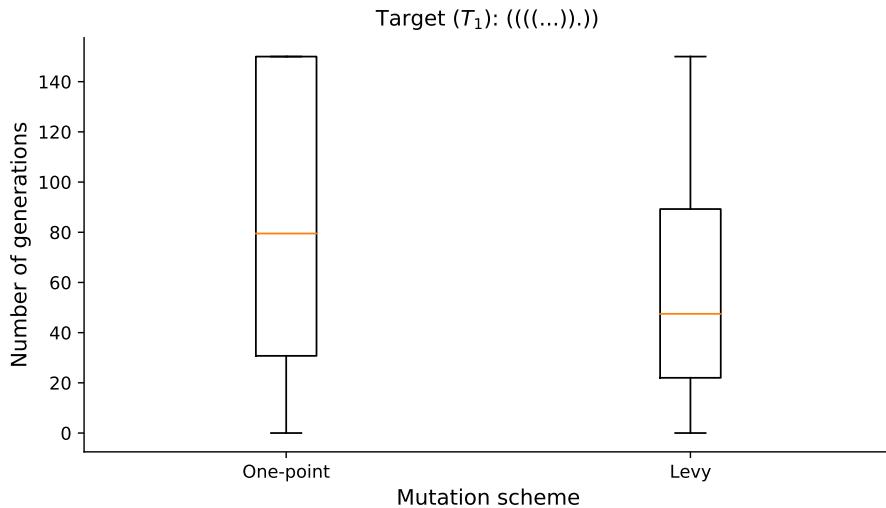


Figure 6.2: Distribution of number of generations need to solve the target T_1 , for both Lévy and Local mutation schemes.

3215 Although we believe that Lévy flight-type search algorithms
 3216 offer a valuable alternative to local search, we emphasise that its
 3217 enhanced performance over say antaRNA is partially influenced by
 3218 the specific capabilities of existing folding tools. Their limitations
 3219 may account for the degradation of these tools as the pseudoknot
 3220 motifs get increasingly complex (i.e. the incapacity of existing
 3221 folding tools to predict some pseudoknot motifs influences the
 3222 performance of both aRNAque and antaRNA). The Lévy mutation
 3223 has also shown less potential in controlling the GC-content of
 3224 the designed sequence when compared to antaRNA on pseudo-
 3225 knotted target structures. antaRNA's parameters used in this work
 3226 were tuned using pKiss; therefore, it could be possible room for
 3227 improving the benchmark presented here by retuning them using
 3228 IPKnot or HotKnots. Another possible limitation is the fact that
 3229 most target structures were relatively easy to solve (in less than
 3230 100 generations), which possibly allowed local search to perform
 3231 better than Lévy search in some cases. Further research on more
 3232 challenging target structures will improve our understanding of
 3233 which conditions favour local *vs.* Lévy search.

3234 6.3 rafft AND EVOLUTIONARY DYNAMICS PERSPECTIVES

3235 The RNA inverse folding has deep connections with theoretical
 3236 evolutionary dynamics studies, where the sequence-secondary
 3237 structure relationship is a popular model for studying the geno-
 3238 type/phenotype maps [59, 78]. Similar to the algorithm imple-
 3239 mented in aRNAque, simulating a dynamic evolutionary process

3240 using RNA sequence-secondary structure relationship as a model
 3241 often involves a population of RNA sequences to a given target
 3242 secondary structure. In such simulation, three main ingredients
 3243 are required: replication, selection and mutation. These are the
 3244 fundamental and defining principles of biological systems. The
 3245 underlying idea is that the genomic material (the blueprint that
 3246 determines the corresponding secondary structure) in the form of
 3247 RNAs is replicated and passed on to the new offspring from gen-
 3248 eration to generation. An RNA individual is then folded into its
 3249 corresponding secondary structure at each generation. Fitness is
 3250 then defined as a function that measures how close is the realized
 3251 structure to the target structure. Therefore, selection results when
 3252 different types of RNA individuals compete with each other. One
 3253 RNA may reproduce faster and thereby out complete the others.
 3254 Occasionally, reproduction involves mistakes; these mistakes are
 3255 termed mutations. Mutations are then responsible for generating
 3256 different RNAs that can be evaluated in the selection process,
 3257 thus resulting in biological novelty and diversity.

3258 Such a simple model gives a unified framework where evolu-
 3259 tionary concepts like plasticity, evolvability, epistasis, neutrality,
 3260 continuity, and modularity can be precisely defined and statisti-
 3261 cally measured [2, 53]. At the molecular level, plasticity is viewed
 3262 as the capacity of an RNA sequence to assume a variety of ener-
 3263 getically favourable secondary structures by equilibrating among
 3264 them at a constant temperature [2]. Such concepts have been
 3265 studied more extensively using the RNA inverse folding as a toy
 3266 model. These studies revealed that selection leads to the reduc-
 3267 tion of plasticity and, therefore, to extreme modularity. Another
 3268 well studied property of evolution is the neutrality which was first
 3269 introduced by Kimura [84], and it suggested that the majority
 3270 of genotypic changes (or mutations) in evolution are selectively
 3271 neutral. The attention to Kimura's contention has led to the dis-
 3272 covery of a neutral network in the context of genotype-phenotype
 3273 models for RNA secondary structure [127, 145]. Two RNA se-
 3274 quences are neutral if they have the same fitness and fold into the
 3275 same RNA secondary structure. Neutrality is a central concept
 3276 in the study of neutral evolution, and many recent studies use
 3277 the sequence-secondary structure relationship as a toy evolution
 3278 model.

3279 An important issue in evolutionary biology concerns the ex-
 3280 tent to which the history of life has proceeded gradually or has
 3281 been punctuated by discontinuous transitions at the level of phe-
 3282 notypes. Distinguishing the notion of continuous from discon-

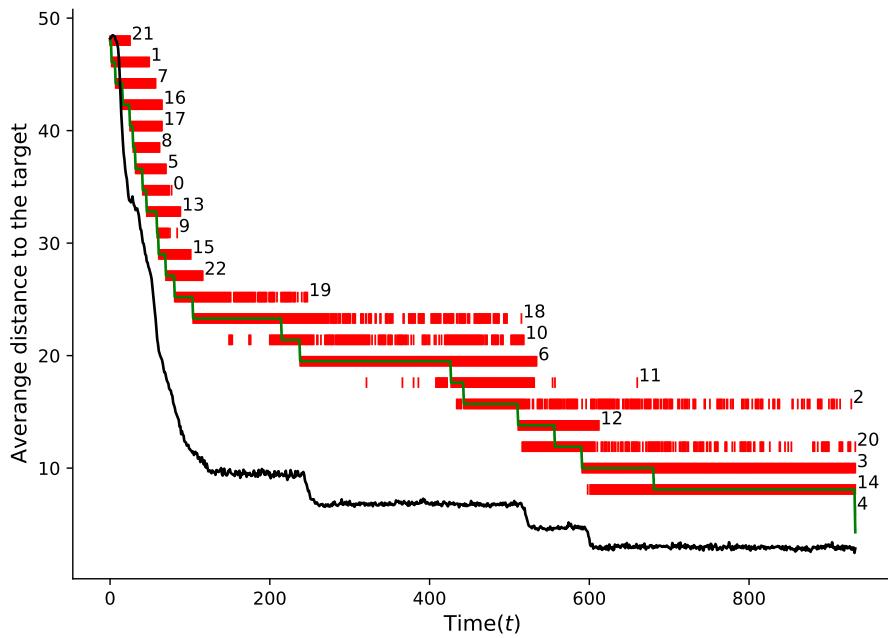
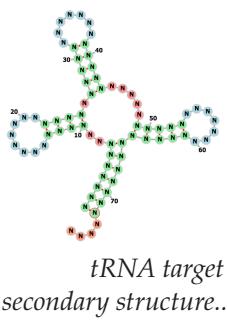


Figure 6.3: Simulation of an RNA population evolving toward a tRNA (See the figure on the right side) target secondary structure. The target was reached after 933 generations (i.e. $\approx 10^5$ replications). The black line shows the average structure distance of the structures in the population to the target structure. The evolutionary history linking the initial structure to the target structure comprises 23. Each structure is labelled by an integer taken from 0 to 22. To each of them corresponds one horizontal line (in red). The top-level corresponds to the initial structure and the bottom the target structure. At each level, a series of red intervals correspond to the periods when the structure was present in the population, and the green curve represents the transition between structures. Only the time axis has a meaning for the red and green curves.

3283 tinuous changes at the level of phenotypes requires a notion of
 3284 nearness between phenotypes. This notion was previously introduced
 3285 by Fontana and Peter [53], and it is based on the probability
 3286 of one phenotype being accessible from another through changes
 3287 in the genotype. The RNA sequence-secondary structure relationship
 3288 provides a framework where the notion of discontinuity
 3289 transition is more precise. It allows understanding of how it arises
 3290 in the model of evolutionary adaptation. This is done by simulating
 3291 an RNA population that evolves toward a tRNA target
 3292 secondary structure in a flow reactor logically constrained to a
 3293 capacity of 1000 sequences. Once the secondary target structure



*tRNA target
secondary structure..*

is found, the evolutionary trajectory is backtraced to identify all the distinct structures involved and the transition between them. Figure 6.3 shows the evolution of the average distance to the tRNA target structure, the intervals of time for which a particular structure is present in the population, and a transition between distinct structures present in the evolutionary path. In Fontana's suggestions, a transition ($S_1 \rightarrow S_2$) between two structures S_1 and S_2 is considered to be continuous if the structure S_1 is 'near' S_2 . In other terms, S_2 is likely to be accessible through the neighbour neutral sequences of S_1 . So if S_2 appears in the evolutionary path at time t , there exists a time $t' < t$ where S_2 was already present in the population. In contrast, the transition is discontinuous otherwise (i.e. the time the structure S_2 appears in the evolutionary path exactly at the same time it was present in the population). An example of continuous transition in Figure 6.3 is the transition $18 \rightarrow 10$ whereas the transition $15 \rightarrow 22$ is said to be discontinuous.

The previous simulation was performed using RNAfold, the folding tool included in the ViennaRNA package. When using ViennaRNA, the plastic ensemble of an RNA sequence ϕ is often considered to be the suboptimal ensemble structure Σ_ϕ within a user-defined energy range above the MFE at a constant temperature T . The ViennaRNA package provides an efficient tools RNAsubopt allowing to compute Σ_ϕ . In a more rigorous implementation of plasticity, each of those structures in the ensemble Σ_ϕ should result from a developmental pathway. Therefore, the environmental changes may induce a change in the developmental path, allowing switching from one structure in the structural ensemble to another. When considering the set of structures produced using RAFFT, each meta-stable structure represents an RNA pathway; therefore, this ensemble can be considered a developmental plastic ensemble. Using RAFFT to simulate the evolutionary dynamic model may provide an alternative framework to study evolutionary concepts like continuity and plasticity. Perhaps, another way of defining continuous transition ($S_1 \rightarrow S_2$) from structure S_1 to S_2 when using will be to check if the structure S_2 is in the RAFFT's structure ensemble of the sequence with MFE S_1 . In that wise, we suggest utilizing RAFFT to study and draw a different interpretation of continuous evolutionary transition.

A

3333

3334 APPENDIX

3335 A.1 arnaque's gc-content parameters

3336 The GC-content is controlled in aRNAque using the mutation pa-
3337 rameters P_C and P_N . The following table gives the corresponding
3338 mutation parameters to the four regimes of GC-content values
3339 used for our benchmark.

Table A.1: Mutation parameters used in aRNAque to control the GC-content values.

GC-content values	P_C	P_N	aRNAque's key
0.25	{0.125, 0.125, 0.3, 0.3, 0.075, 0.075}	{0.125, 0.125, 0.375, 0.375}	GC25
0.25	{0.25, 0.25, 0.2, 0.2, 0.05, 0.05}	{0.25, 0.25, 0.25, 0.5}	GC50
0.75	{0.375, 0.375, 0.1, 0.1, 0.025, 0.025}	{0.375, 0.375, 0.125, 0.125}	GC75
1.0	{0.5, 0.5, 0.0, 0.0, 0.0, 0.0}	{0.5, 0.5, 0., 0.}	GC

3340 A.2 BENCHMARK ON eternal100 DATASET

3341 For each of the benchmarks on the Eterna100 datasets, We ran
3342 the first benchmark using the default aRNAque's parameter config-
3343 uration. And then, the unsolved structures are sorted out to run
3344 a second benchmark with a maximum number of generations
3345 set at 5000. aRNAque's performance presented in the paper is a
3346 combination of all the designed sequences for each realisation.

3347 A.3 GENERAL EA BENCHMARK PARAMETERS

3348 The same hardware resources and the same computer are used for
3349 all the benchmarks listed in the following table. A supercomputer
3350 with 40-Core Intel Xeon E5-2698 v4 at 2.2 GHz and 512 GB of
3351 RAM with a Debian OS.

Table A.2: Evolutionary algorithm parameter for each benchmarks.

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

3352 A.4 OTHER BENCHMARK ON eterna100-v1

3353 The results on Eterna100-V1 presented in the paper are the best
 3354 of all the benchmarks we have performed. Since our mutation
 3355 scheme relies on the nucleotide distributions which implicitly
 3356 control the GC-content of the designed sequences, to obtain our
 3357 results, we first selected an arbitrary set of pairs $\{P_N, P_C\}$ and
 3358 benchmark aRNAque on Eterna100-V1 for each of them. The suc-
 3359 cess rate measures the fraction of sequences successfully folding
 3360 into the target structure. Table A.3 shows the different parameters
 3361 we considered and the corresponded input key parameter using
 3362 the call of aRNAque script. Summary of the benchmark presented
 3363 in Table A.4 is obtained by launching for each target structure 5
 3364 independent runs, with a population size of 100 and a maximum
 3365 number of generations of 5000. The energy parameter used here
 3366 was the Turner1999. The dashes in the table mean the benchmarks
 3367 have not been performed for the parameters.

Table A.3: Different parameters for the base pair distributions

Key	$P_N = \{p_A, p_G, p_U, p_C\}$	$P_C = \{p_{GC}, p_{CG}, p_{AU}, p_{UA}, p_{GU}, p_{UG}\}$
ALL	$P_N = \{0.25, 0.25, 0.25, 0.25\}$	$P_C = \{0.2, 0.2, 0.1, 0.1, 0.2, 0.2\}$
GC	$P_N = \{0.25, 0.25, 0.25, 0.25\}$	$P_C = \{0.5, 0.5, 0, 0, 0, 0\}$
GC ₁	$P_N = \{0.25, 0.65, 0.05, 0.05\}$	$P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$
GC ₂	$P_N = \{0.7, 0.1, 0.1, 0.1\}$	$P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$
GC ₃	$P_N = \{0.75, 0.1, 0.1, 0.05\}$	$P_C = \{0.4, 0.5, 0.1, 0, 0, 0\}$
GC ₄	$P_N = \{0.95, 0, 0.05, 0\}$	$P_C = \{0.4, 0.4, 0.2, 0, 0, 0\}$
GC ₅	$P_N = \{0.7, 0.1, 0.1, 0.1\}$	$P_C = \{0.3, 0.2, 0.2, 0.1, 0.1, 0.1\}$

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

Tools	BP param	Mutation param	Percentage of success	$\#(Med(gen_{Zipf}) < Med(gen_{op}))$	$\#(Med(gen_{Zipf}) > Med(gen_{op}))$
aRNAque	<i>ALL</i>	Zipf ($c = 1$)	67%	7(#4)	
		One point	81%	64(#4)	
aRNAque	<i>GC</i>	Zipf ($c = 1$)	80%	43(#10)	
		One point	90%	30(#474)	
aRNAque	<i>GC₁</i>	Zipf ($c = 1$)	84%	29(#4)	
		One point	90%	33(#7)	
aRNAque	<i>GC₂</i>	Zipf ($c = 1$)	89%	61(#10)	
		One point	91%	19(#1920)	
aRNAque	<i>GC₃</i>	Zipf ($c = 1$)	88%	—	
		One point	—	—	
aRNAque	<i>GC₄</i>	Zipf ($c = 1$)	—	—	
		One point	—	—	
aRNAque	<i>GC₅</i>	Zipf ($c = 1$)	82%	44(#9)	
		One point	83%	30(#145)	
Total	—	Zipf ($c = 1$)	90%		
		One point	92%		
		RNAinverse	87%		

3368 A.5 TOOLS PATCHING

3369 To be able to perform our benchmarks, some slight modifications
 3370 was made on **HotKnots** and **antaRNA**. Details about the modifica-
 3371 tions are provided in this section.

- 3372 • **antaRNA**: The change was made at the line 1178 column 7,
 3373 where the line `args = 'HotKnots -m CC -s' + sequence` was
 3374 replaced by to `args = './HotKnots -m CC -s' + sequence`.
- 3375 • **HotKnots**: to run **HotKnots**, we have to move **aRNAque** to the
 3376 bin directory. To avoid that, we updated the source code
 3377 and recompiled a new bin that does not require to move
 3378 **aRNAque** to the bin directory of **HotKnots**. We have uploaded
 3379 the patched version of **HotKnots** in a third-part folder in
 3380 **aRNAque**'s repository for benchmark reproduction.
- 3381 **NB**: The patches do not affect the folding algorithm. It con-
 3382 sisted of avoiding the use of relative paths in **HotKnots**.

3383 A.6 arnaque EXAMPLE CALLS

3384 aRNAque computes the RNA inverse folding problem for different
 3385 classes of structure complexities.

3386 For a pseudo-knot free target secondary structure:

3387

```
3388 $ python aRNAque.py -t "((....)).((....))"
  -bp "GC2"
  -sm "NED"
  -ft "v"
  --job 5
```

3393

3394 Here,

3395 A result to this call could look like this:

3396

```
3397 GCUACGGCACCGUCAGG | ((....)).((....)) | -2.8 | 1.0
  3398 GGGGGACCACCGGUGGG | ((....)).((....)) | -2.5 | 1.0
  3399 GGGCCACCAGCGAAAGC | ((....)).((....)) | -2.2 | 1.0
  3400 GGAAAUCCACCGGAAGG | ((....)).((....)) | -1.4 | 1.0
  3401 GCAAGAGCGCCGCAAGG | ((....)).((....)) | -1.2 | 1.0
```

3402

3403 Where the columns shows respectively the designed sequences,
 3404 the MFE structures, their free energy and the fitness to the target
 3405 (See Equation 5.1)

[June 17, 2022 at 17:55 – 1.0]

- 3407 [1] Fabian Amman, Stephan H. Bernhart, Gero Doose, Ivo L.
3408 Hofacker, Jing Qin, Peter F. Stadler, and Sebastian Will.
3409 “The trouble with long-range base pairs in RNA folding.”
3410 In: *Advances in Bioinformatics and Computational Biology*.
3411 Advances in Bioinformatics and Computational Biology.
3412 Springer International Publishing, 2013, pp. 1–11. doi:
3413 [10.1007/978-3-319-02624-4_1](https://doi.org/10.1007/978-3-319-02624-4_1). URL: https://doi.org/10.1007/978-3-319-02624-4_1.
- 3415 [2] Lauren W Ancel and Walter Fontana. “Plasticity, evolv-
3416 ability, and modularity in RNA.” In: *Journal of Experi-
3417 mental Zoology* 288.3 (2000), pp. 242–283.
- 3418 [3] Jeff Anderson-Lee, Eli Fisker, Vineet Kosaraju, Michelle
3419 Wu, Justin Kong, Jeehyung Lee, Minjae Lee, Mathew
3420 Zada, Adrien Treuille, and Rhiju Das. “Principles for pre-
3421 dicting RNA secondary structure design difficulty.” In: *Journal of molecular biology* 428.5 (2016), pp. 748–757.
- 3423 [4] Mirela Andronescu, Vera Bereg, Holger H Hoos, and
3424 Anne Condon. “RNA STRAND: the RNA secondary struc-
3425 ture and statistical analysis database.” In: *BMC Bioinfor-
3426 matics* 9.1 (2008), pp. 1–10.
- 3427 [5] Mirela Andronescu, Anthony P Fejes, Frank Hutter, Hol-
3428 ger H Hoos, and Anne Condon. “A new algorithm for
3429 RNA secondary structure design.” In: *Journal of molecular
3430 biology* 336.3 (2004), pp. 607–624.
- 3431 [6] Assaf Avihoo, Alexander Churkin, and Danny Barash.
3432 “RNAexinv: An extended inverse RNA folding from shape
3433 and physical attributes to sequences.” In: *BMC bioinfor-
3434 matics* 12.1 (2011), pp. 1–8.
- 3435 [7] Rolf Backofen and Wolfgang R Hess. “Computational
3436 prediction of sRNAs and their targets in bacteria.” In:
3437 *RNA biology* 7.1 (2010), pp. 33–42.
- 3438 [8] Rodolphe Barrangou, Christophe Fremaux, Hélène De-
3439 veau, Melissa Richards, Patrick Boyaval, Sylvain Moineau,
3440 Dennis A Romero, and Philippe Horvath. “CRISPR pro-
3441 vides acquired resistance against viruses in prokaryotes.”
3442 In: *Science* 315.5819 (2007), pp. 1709–1712.

- [9] S. Bellaousov and D. H. Mathews. "Probknot: fast prediction of RNA secondary structure including pseudoknots." In: *RNA* 16.10 (2010), pp. 1870–1880. doi: 10.1261/rna.2125310. URL: <https://doi.org/10.1261/rna.2125310>.
- [10] Jan H Bergmann and David L Spector. "Long non-coding RNAs: modulators of nuclear structure and function." In: *Current opinion in cell biology* 26 (2014), pp. 10–18.
- [11] Philip C Bevilacqua, Laura E Ritchey, Zhao Su, and Sarah M Assmann. "Genome-wide analysis of RNA secondary structure." In: *Annual review of genetics* 50 (2016), pp. 235–266.
- [12] Eckart Bindewald, Kirill Afonin, Luc Jaeger, and Bruce A Shapiro. "Multistrand RNA secondary structure prediction and nanostructure design including pseudoknots." In: *ACS nano* 5.12 (2011), pp. 9542–9551.
- [13] Édouard Bonnet, Paweł Rzążewski, and Florian Sikora. "Designing RNA secondary structures is hard." In: *Journal of Computational Biology* 27.3 (2020), pp. 302–316.
- [14] Grégoire Bonnet, Oleg Krichevsky, and Albert Libchaber. "Kinetics of conformational fluctuations in DNA hairpin-loops." In: *Proceedings of the National Academy of Sciences* 95.15 (1998), pp. 8602–8606.
- [15] Ronald R Breaker, RF Gesteland, TR Cech, and JF Atkins. *The RNA world*. Cold Spring Harbor Laboratory Press, New York, 2006.
- [16] Philippe Brion and Eric Westhof. "Hierarchy and dynamics of RNA folding." In: *Annual review of biophysics and biomolecular structure* 26.1 (1997), pp. 113–137.
- [17] James W Brown. "The ribonuclease P database." In: *Nucleic Acids Research* 26.1 (1998), pp. 351–352.
- [18] Anke Busch and Rolf Backofen. "INFO-RNA—a fast approach to inverse RNA folding." In: *Bioinformatics* 22.15 (2006), pp. 1823–1831.
- [19] Thomas R Cech and Joan A Steitz. "The noncoding RNA revolution—trashing old rules to forge new ones." In: *Cell* 157.1 (2014), pp. 77–94.

- [20] Shaon Chakrabarti, Changbong Hyeon, Xiang Ye, George H Lorimer, and D Thirumalai. "Molecular chaperones maximize the native state yield on biological times by driving substrates out of equilibrium." In: *Proceedings of the National Academy of Sciences* 114.51 (2017), E10919–E10927.
- [21] James Chappell, Kyle E Watters, Melissa K Takahashi, and Julius B Lucks. "A renaissance in RNA synthetic biology: new mechanisms, applications and tools for the future." In: *Current opinion in chemical biology* 28 (2015), pp. 47–56.
- [22] Shi-Jie Chen. "RNA folding: conformational statistics, folding kinetics, and ion electrostatics." In: *Annu. Rev. Biophys.* 37 (2008), pp. 197–214.
- [23] Alexander Churkin, Matan Drory Retwitzer, Vladimir Reinhartz, Yann Ponty, Jérôme Waldspühl, and Danny Barash. "Design of RNAs: comparing programs for inverse RNA folding." In: *Briefings in bioinformatics* 19.2 (2017), pp. 350–358.
- [24] Simona Cocco, John F Marko, and Remi Monasson. "Slow nucleic acid unzipping kinetics from sequence-defined barriers." In: *The European Physical Journal E* 10.2 (2003), pp. 153–161.
- [25] Francis Crick. "Central dogma of molecular biology." In: *Nature* 227.5258 (1970), pp. 561–563.
- [26] Emilio Cusanelli and Pascal Chartrand. "Telomeric non-coding RNA: telomeric repeat-containing RNA in telomere biology." In: *Wiley Interdisciplinary Reviews: RNA* 5.3 (2014), pp. 407–419.
- [27] Paul Dallaire and François Major. "Exploring alternative RNA structure sets using MC-flashfold and db2cm." In: *RNA Structure Determination*. Springer, 2016, pp. 237–251.
- [28] Simon H Damberger and Robin R Gutell. "A comparative database of group I intron structures." In: *Nucleic Acids Research* 22.17 (1994), pp. 3508–3510.
- [29] Christian Darabos, Mario Giacobini, Ting Hu, and Jason H. Moore. "Lévy-Flight Genetic Programming: Towards a New Mutation Paradigm." In: *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Ed. by Mario Giacobini, Leonardo Vanneschi, and William

- 3519 S. Bush. Berlin, Heidelberg: Springer Berlin Heidelberg,
 3520 2012, pp. 38–49. ISBN: 978-3-642-29066-4.
- 3521 [30] Kévin Darty, Alain Denise, and Yann Ponty. “VARNA:
 3522 Interactive drawing and editing of the RNA secondary
 3523 structure.” In: *Bioinformatics* 25.15 (2009), p. 1974.
- 3524 [31] Jennifer Daub, Paul P Gardner, John Tate, Daniel Ram-
 3525 sköld, Magnus Manske, William G Scott, Zasha Wein-
 3526 berg, Sam Griffiths-Jones, and Alex Bateman. “The RNA
 3527 WikiProject: community annotation of RNA families.” In:
 3528 *RNA* 14.12 (2008), pp. 2462–2464.
- 3529 [32] Christoph Dieterich and Peter F Stadler. “Computational
 3530 biology of RNA interactions.” In: *Wiley Interdisciplinary
 3531 Reviews: RNA* 4.1 (2013), pp. 107–120.
- 3532 [33] Ken A Dill. “Additivity principles in biochemistry.” In:
 3533 *Journal of Biological Chemistry* 272.2 (1997), pp. 701–704.
- 3534 [34] Robert M. Dirks and Niles A. Pierce. “A partition function
 3535 algorithm for nucleic acid secondary structure includ-
 3536 ing pseudoknots.” In: *Journal of Computational Chemistry*
 3537 24.13 (2003). _eprint: <https://onlinelibrary.wiley.com/-doi/pdf/10.1002/jcc.10296>, pp. 1664–1677. ISSN: 1096-987X.
 3538 DOI: <https://doi.org/10.1002/jcc.10296>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.10296> (visited on 02/17/2021).
- 3542 [35] Robert M Dirks and Niles A Pierce. “A partition function
 3543 algorithm for nucleic acid secondary structure including
 3544 pseudoknots.” In: *Journal of computational chemistry* 24.13
 3545 (2003), pp. 1664–1677.
- 3546 [36] Chuong B Do, Daniel A Woods, and Serafim Batzoglou.
 3547 “CONTRAFold: RNA secondary structure prediction with-
 3548 out physics-based models.” In: *Bioinformatics* 22.14 (2006),
 3549 e90–e98.
- 3550 [37] Elizabeth A Doherty and Jennifer A Doudna. “Ribozyme
 3551 structures and mechanisms.” In: *Annual Review of Bio-
 3552 physics and Biomolecular Structure* 30.1 (2001), pp. 457–
 3553 475.
- 3554 [38] Ivan Dotu, Juan Antonio Garcia-Martin, Betty L Slinger,
 3555 Vinodh Mechery, Michelle M Meyer, and Peter Clote.
 3556 “Complete RNA inverse folding: computational design
 3557 of functional hammerhead ribozymes.” In: *Nucleic acids
 3558 research* 42.18 (2014), pp. 11752–11762.

- 3559 [39] Robin D Dowell and Sean R Eddy. "Evaluation of several
 3560 lightweight stochastic context-free grammars for RNA
 3561 secondary structure prediction." In: *BMC bioinformatics*
 3562 5.1 (2004), pp. 1–14.
- 3563 [40] N Dromi, A Avihoo, and D Barash. "Reconstruction of nat-
 3564 ural RNA sequences from RNA shape, thermodynamic
 3565 stability, mutational robustness, and linguistic complexity
 3566 by evolutionary computation." In: *Journal of Biomolecular*
 3567 *Structure and Dynamics* 26.1 (2008), pp. 147–161.
- 3568 [41] Matan Drory Retwitzer, Vladimir Reinhartz, Yann Ponty,
 3569 Jérôme Waldspühl, and Danny Barash. "incaRNABinv:
 3570 a web server for the fragment-based design of RNA se-
 3571 quences." In: *Nucleic acids research* 44.W1 (2016), W308–
 3572 W314.
- 3573 [42] Andrew D Ellington, Xi Chen, Michael Robertson, and
 3574 Angel Syrett. "Evolutionary origins and directed evolu-
 3575 tion of RNA." In: *The international journal of biochemistry*
 3576 & cell biology 41.2 (2009), pp. 254–265.
- 3577 [43] Ali Esmaili-Taheri and Mohammad Ganjtabesh. "ERD:
 3578 a fast and reliable tool for RNA design including con-
 3579 straints." In: *BMC bioinformatics* 16.1 (2015), p. 20.
- 3580 [44] Ali Esmaili-Taheri, Mohammad Ganjtabesh, and Morteza
 3581 Mohammad-Noori. "Evolutionary solution for the RNA
 3582 design problem." In: *Bioinformatics* 30.9 (2014), pp. 1250–
 3583 1258.
- 3584 [45] Manel Esteller. "Non-coding RNAs in human disease."
 3585 In: *Nature reviews genetics* 12.12 (2011), pp. 861–874.
- 3586 [46] Jörg Fallmann, Sebastian Will, Jan Engelhardt, Björn Grün-
 3587 ning, Rolf Backofen, and Peter F Stadler. "Recent advances
 3588 in RNA folding." In: *Journal of Biotechnology* 261 (2017),
 3589 pp. 97–104.
- 3590 [47] Alessandro Fatica and Irene Bozzoni. "Long non-coding
 3591 RNAs: new players in cell differentiation and develop-
 3592 ment." In: *Nature Reviews Genetics* 15.1 (2014), pp. 7–21.
- 3593 [48] Sven Findeiß, Manja Wachsmuth, Mario Mörl, and Peter F
 3594 Stadler. "Design of transcription regulating riboswitches."
 3595 In: *Methods in enzymology*. Vol. 550. Elsevier, 2015, pp. 1–
 3596 22.

- 3597 [49] Christoph Flamm, Walter Fontana, Ivo L Hofacker, and
 3598 Peter Schuster. "RNA folding at elementary step resolu-
 3599 tion." In: *Rna* 6.3 (2000), pp. 325–338.
- 3600 [50] Christoph Flamm, Ivo L Hofacker, Sebastian Maurer-
 3601 Stroh, Peter F Stadler, and Martin Zehl. "Design of multi-
 3602 stable RNA molecules." In: *Rna* 7.2 (2001), pp. 254–265.
- 3603 [51] Christoph Flamm, Ivo L. Hofacker, Peter F. Stadler, and
 3604 Michael T. Wolfinger. "Barrier trees of degenerate land-
 3605 scapes." In: *Zeitschrift für Physikalische Chemie* 216.2 (2002),
 3606 nil. doi: 10.1524/zpch.2002.216.2.155. URL: <https://doi.org/10.1524/zpch.2002.216.2.155>.
- 3608 [52] Peter J Flor, James B Flanagan, and TR Cech. "A con-
 3609 served base pair within helix P4 of the Tetrahymena ri-
 3610 bozyme helps to form the tertiary structure required for
 3611 self-splicing." In: *The EMBO Journal* 8.11 (1989), pp. 3391–
 3612 3399.
- 3613 [53] Walter Fontana and Peter Schuster. "Continuity in evo-
 3614 lution: on the nature of transitions." In: *Science* 280.5368
 3615 (1998), pp. 1451–1455.
- 3616 [54] Jacques R. Fresco, Bruce M. Alberts, and Paul Doty. "Some
 3617 Molecular Details of the Secondary Structure of Ribonu-
 3618 cleic Acid." In: *Nature* 188.4745 (Oct. 1960). Number:
 3619 4745 Publisher: Nature Publishing Group, pp. 98–101.
 3620 ISSN: 1476-4687. doi: 10.1038/188098a0. URL: <https://www.nature.com/articles/188098a0> (visited on
 3621 04/14/2021).
- 3623 [55] James ZM Gao, Linda YM Li, and Christian M Reidys.
 3624 "Inverse folding of RNA pseudoknot structures." In: *Al-
 3625 gorithms for Molecular Biology* 5.1 (2010), pp. 1–19.
- 3626 [56] Juan Antonio Garcia-Martin, Peter Clote, and Ivan Dotu.
 3627 "RNAAiFOLD: a constraint programming algorithm for
 3628 RNA inverse folding and molecular design." In: *Journal
 3629 of bioinformatics and computational biology* 11.02 (2013),
 3630 p. 1350001.
- 3631 [57] Paul P Gardner, Jennifer Daub, John G Tate, Eric P Nawrocki,
 3632 Diana L Kolbe, Stinus Lindgreen, Adam C Wilkinson,
 3633 Robert D Finn, Sam Griffiths-Jones, Sean R Eddy, et al.
 3634 "Rfam: updates to the RNA families database." In: *Nucleic
 3635 Acids Research* 37.suppl_1 (2009), pp. D136–D140.

- 3636 [58] Michael Geis, Christoph Flamm, Michael T Wolfinger, An-
 3637 drea Tanzer, Ivo L Hofacker, Martin Middendorf, Chris-
 3638 tian Mandl, Peter F Stadler, and Caroline Thurner. "Fold-
 3639 ing kinetics of large RNAs." In: *Journal of Molecular Biology*
 3640 379.1 (2008), pp. 160–173.
- 3641 [59] Sam F Greenbury, Steffen Schaper, Sebastian E Ahnert,
 3642 and Ard A Louis. "Genetic correlations greatly increase
 3643 mutational robustness and can both reduce and enhance
 3644 evolvability." In: *PLoS computational biology* 12.3 (2016),
 3645 e1004773.
- 3646 [60] Peixuan Guo. "The emerging field of RNA nanotechnol-
 3647 ogy." In: *Nature nanotechnology* 5.12 (2010), pp. 833–842.
- 3648 [61] Zhuyan Guo and D Thirumalai. "Kinetics of protein fold-
 3649 ing: nucleation mechanism, time scales, and pathways." In:
 3650 *Biopolymers: Original Research on Biomolecules* 36.1 (1995),
 3651 pp. 83–102.
- 3652 [62] Robin R Gutell. "Collection of small subunit (16S-and
 3653 16S-like) ribosomal RNA structures: 1994." In: *Nucleic
 3654 Acids Research* 22.17 (1994), pp. 3502–3507.
- 3655 [63] Robin R Gutell, Michael W Gray, and Murray N Schnare.
 3656 "A compilation of large subunit (23S and 23S-like) ribo-
 3657 somal RNA structures: 1993." In: *Nucleic Acids Research*
 3658 21.13 (1993), p. 3055.
- 3659 [64] Robin R Gutell, Jung C Lee, and Jamie J Cannone. "The
 3660 accuracy of ribosomal RNA comparative structure mod-
 3661 els." In: *Current opinion in structural biology* 12.3 (2002),
 3662 pp. 301–310.
- 3663 [65] Robin R Gutell, Bryn Weiser, Carl R Woese, and Harry
 3664 F Noller. "Comparative anatomy of 16-S-like ribosomal
 3665 RNA." In: *Progress in nucleic acid research and molecular
 3666 biology* 32 (1985), pp. 155–216.
- 3667 [66] Christine E Hajdin, Stanislav Bellaousov, Wayne Huggins,
 3668 Christopher W Leonard, David H Mathews, and Kevin M
 3669 Weeks. "Accurate SHAPE-directed RNA secondary struc-
 3670 ture modeling, including pseudoknots." In: *Proceedings of
 3671 the National Academy of Sciences* 110.14 (2013), pp. 5498–
 3672 5503.

- 3673 [67] Arif Ozgun Harmanci, Gaurav Sharma, and David H
 3674 Mathews. "Stochastic sampling of the RNA structural
 3675 alignment space." In: *Nucleic acids research* 37.12 (2009),
 3676 pp. 4063–4075.
- 3677 [68] Teresa Haynes, Debra Knisley, and Jeff Knisley. "Using
 3678 a neural network to identify secondary RNA structures
 3679 quantified by graphical invariants." In: *Comm Math Com-*
 3680 *put Chem* 60 (2008), pp. 277–290.
- 3681 [69] Ivo L Hofacker, Walter Fontana, Peter F Stadler, L Sebas-
 3682 tian Bonhoeffer, Manfred Tacker, and Peter Schuster. "Fast
 3683 folding and comparison of RNA secondary structures." In:
 3684 *Monatshefte für Chemie/Chemical Monthly* 125.2 (1994),
 3685 pp. 167–188.
- 3686 [70] Ivo L. Hofacker, Peter F. Stadler, and Peter F. Stadler.
 3687 "RNA Secondary Structures." In: *Reviews in Cell Biology*
 3688 and *Molecular Medicine*. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/3527600906.mcb.200500009>. Amer-
 3689 ican Cancer Society, 2006. ISBN: 978-3-527-60090-8. doi:
 3690 [10.1002/3527600906.mcb.200500009](https://doi.org/10.1002/3527600906.mcb.200500009). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/3527600906.mcb.200500009> (visited on 03/03/2021).
- 3694 [71] J Holland. *Adaptation in Natural and Artificial Systems*, Ann
 3695 Arbor: University of Michigan Press, 1975. 1992.
- 3696 [72] Chiou-Yi Hor, Chang-Biau Yang, Chia-Hung Chang, Chiou-
 3697 Ting Tseng, and Hung-Hsin Chen. "A Tool preference
 3698 choice Method for RnA secondary structure prediction by
 3699 sVM with statistical Tests." In: *Evolutionary Bioinformatics*
 3700 9 (2013), EBO-S10580.
- 3701 [73] Liang Huang, He Zhang, Dezhong Deng, Kai Zhao, Kaibo
 3702 Liu, David A Hendrix, and David H Mathews. "Lin-
 3703 earFold: linear-time approximate RNA folding by 5'-to-
 3704 3'dynamic programming and beam search." In: *Bioinfor-*
 3705 *matics* 35.14 (2019), pp. i295–i304.
- 3706 [74] Woong Y Hwang, Yanfang Fu, Deepak Reyon, Morgan
 3707 L Maeder, Shengdar Q Tsai, Jeffry D Sander, Randall T
 3708 Peterson, J-R Joanna Yeh, and J Keith Joung. "Efficient in
 3709 vivo genome editing using RNA-guided nucleases." In:
 3710 *Nature biotechnology* 31.3 (2013), p. 227.
- 3711 [75] Farren J Isaacs, Daniel J Dwyer, and James J Collins. "RNA
 3712 synthetic biology." In: *Nature biotechnology* 24.5 (2006),
 3713 pp. 545–554.

- [76] Hervé Isambert and Eric D Siggia. "Modeling RNA folding paths with pseudoknots: application to hepatitis delta virus ribozyme." In: *Proceedings of the National Academy of Sciences* 97.12 (2000), pp. 6515–6520.
- [77] Tor Ivry, Shahar Michal, Assaf Avihoo, Guillermo Sapiro, and Danny Barash. "An image processing approach to computing distances between RNA secondary structures dot plots." In: *Algorithms for Molecular Biology* 4.1 (2009), pp. 1–19.
- [78] Luc Jaeger, Eric Westhof, and Neocles B Leontis. "TectoRNA: modular assembly units for the construction of RNA nano-objects." In: *Nucleic acids research* 29.2 (2001), pp. 455–463.
- [79] Stefan Janssen and Robert Giegerich. "The RNA shapes studio." In: *Bioinformatics* (2015). doi: [10.1093/bioinformatics/btu649](https://doi.org/10.1093/bioinformatics/btu649). URL: <http://bioinformatics.oxfordjournals.org/content/31/3/423.abstract>.
- [80] Martin Jinek, Alexandra East, Aaron Cheng, Steven Lin, Enbo Ma, and Jennifer Doudna. "RNA-programmed genome editing in human cells." In: *elife* 2 (2013), e00471.
- [81] Anis Farhan Kamaruzaman, Azlan Mohd Zain, Suhaila Mohamed Yusuf, and Amirmudin Udin. "Lévy flight algorithm for optimization problems—a literature review." In: *Applied Mechanics and Materials*. Vol. 421. Trans Tech Publ. 2013, pp. 496–501.
- [82] Michael Kertesz, Yue Wan, Elad Mazor, John L Rinn, Robert C Nutter, Howard Y Chang, and Eran Segal. "Genome-wide measurement of RNA secondary structure in yeast." In: *Nature* 467.7311 (2010), pp. 103–107.
- [83] Yoon Ki Kim, Luc Furic, Marc Parisien, François Major, Luc DesGroseillers, and Lynne E Maquat. "Staufen1 regulates diverse classes of mammalian transcripts." In: *The EMBO journal* 26.11 (2007), pp. 2670–2681.
- [84] Motoo Kimura. *The neutral theory of molecular evolution*. Cambridge University Press, 1983.
- [85] Robert Kleinkauf, Torsten Houwaart, Rolf Backofen, and Martin Mann. "antaRNA—Multi-objective inverse folding of pseudoknot RNA using ant-colony optimization." In: *BMC bioinformatics* 16.1 (2015), pp. 1–7.

- [86] Robert Kleinkauf, Martin Mann, and Rolf Backofen. “antaRNA: ant colony-based RNA sequence design.” In: *Bioinformatics* 31.19 (2015), pp. 3114–3121.
- [87] Konstantin Klemm, Christoph Flamm, and Peter F Stadler. “Funnels in energy landscapes.” In: *The European Physical Journal B* 63.3 (2008), pp. 387–391.
- [88] Bjarne Knudsen and Jotun Hein. “RNA secondary structure prediction using stochastic context-free grammars and evolutionary history.” In: *Bioinformatics (Oxford, England)* 15.6 (1999), pp. 446–454.
- [89] Bjarne Knudsen and Jotun Hein. “Pfold: RNA secondary structure prediction using stochastic context-free grammars.” In: *Nucleic acids research* 31.13 (2003), pp. 3423–3428.
- [90] Donald E. Knuth. “Computer Programming as an Art.” In: *Communications of the ACM* 17.12 (1974), pp. 667–673.
- [91] Rohan V Koodli, Benjamin Keep, Katherine R Coppess, Fernando Portela, Eterna participants, and Rhiju Das. “EternaBrain: Automated RNA design through move sets and strategies from an Internet-scale RNA videogame.” In: *PLoS computational biology* 15.6 (2019), e1007059.
- [92] Rohan V. Koodli, Boris Rudolfs, Hannah K. Wayment-Steele, Eterna Structure Designers, and Rhiju Das. “Re-designing the EteRNA100 for the Vienna 2 folding engine.” In: *bioRxiv* (2021). doi: 10.1101/2021.08.26.457839. eprint: [https://www.biorxiv.org/content/early/2021/08/28/2021.08.26.457839](https://www.biorxiv.org/content/early/2021/08/28/2021.08.26.457839.full.pdf). URL: <https://www.biorxiv.org/content/early/2021/08/28/2021.08.26.457839>.
- [93] Yingjun Li, Saifu Pan, Yan Zhang, Min Ren, Mingxia Feng, Nan Peng, Lanming Chen, Yun Xiang Liang, and Qunxin She. “Harnessing Type I and Type III CRISPR-Cas systems for genome editing.” In: *Nucleic acids research* 44.4 (2016), e34–e34.
- [94] Zhongsen Li, Zhan-Bin Liu, Aiqiu Xing, Bryan P Moon, Jessica P Koellhoffer, Lingxia Huang, R Timothy Ward, Elizabeth Clifton, S Carl Falco, and A Mark Cigan. “Cas9-guide RNA directed genome editing in soybean.” In: *Plant physiology* 169.2 (2015), pp. 960–970.

- [95] Adam Lipowski and Dorota Lipowska. "Roulette-wheel selection via stochastic acceptance." In: *Physica A: Statistical Mechanics and its Applications* 391.6 (2012), pp. 2193–2196.
- [96] Qi Liu, Xiuzi Ye, and Yin Zhang. "A Hopfield neural network based algorithm for RNA secondary structure prediction." In: *First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*. Vol. 1. IEEE. 2006, pp. 10–16.
- [97] Ronny Lorenz, Stephan H Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. "Viennarna Package 2.0." In: *Algorithms for Molecular Biology* 6.1 (2011), p. 26. doi: [10.1186/1748-7188-6-26](https://doi.org/10.1186/1748-7188-6-26). URL: <https://doi.org/10.1186/1748-7188-6-26>.
- [98] Ronny Lorenz, Stephan H Bernhart, Christian Höner Zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. "ViennaRNA Package 2.0." In: *Algorithms for molecular biology* 6.1 (2011), p. 26.
- [99] Ronny Lorenz, Christoph Flamm, Ivo Hofacker, and Peter Stadler. "Efficient computation of base-pairing probabilities in multi-strand RNA folding." In: *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies*. 2020, pp. 23–31. doi: [10.5220/0008916600230031](https://doi.org/10.5220/0008916600230031). URL: <https://doi.org/10.5220/0008916600230031>.
- [100] Zhi John Lu, Jason W Gloor, and David H Mathews. "Improved RNA secondary structure prediction by maximizing expected pair accuracy." In: *Rna* 15.10 (2009), pp. 1805–1813.
- [101] Rune B Lyngsø, James WJ Anderson, Elena Sizikova, Amarendra Badugu, Tomas Hyland, and Jotun Hein. "Fr-nakenstein: multiple target inverse RNA folding." In: *BMC bioinformatics* 13.1 (2012), pp. 1–12.
- [102] JT Madison, GA Everett, and H Kung. "Nucleotide sequence of a yeast tyrosine transfer RNA." In: *Science* 153.3735 (1966), pp. 531–534.
- [103] B Mandelbrot. "Certain speculative prices (1963)." In: *The Journal of Business* 45.4 (1972), pp. 542–543.

- 3831 [104] Hugo M. Martinez. "An RNA folding rule." In: *Nucleic
3832 Acids Research* 12.1 (1984), pp. 323–334. doi: [10.1093/nar/12.1part1.323](https://doi.org/10.1093/nar/12.1part1.323). URL: <https://doi.org/10.1093/nar/12.1part1.323>.
- 3835 [105] David H. Mathews. "How to benchmark RNA secondary
3836 structure prediction accuracy." In: *Methods* 162–163.162
3837 (2019), pp. 60–67. doi: [10.1016/j.ymeth.2019.04.003](https://doi.org/10.1016/j.ymeth.2019.04.003).
3838 URL: <https://doi.org/10.1016/j.ymeth.2019.04.003>.
- 3839 [106] David H Mathews, Matthew D Disney, Jessica L Childs,
3840 Susan J Schroeder, Michael Zuker, and Douglas H Turner.
3841 "Incorporating chemical modification constraints into a
3842 dynamic programming algorithm for prediction of RNA
3843 secondary structure." In: *Proceedings of the National Academy
3844 of Sciences* 101.19 (2004), pp. 7287–7292.
- 3845 [107] David H Mathews, Jeffrey Sabina, Michael Zuker, and
3846 Douglas H Turner. "Expanded sequence dependence of
3847 thermodynamic parameters improves prediction of RNA
3848 secondary structure." In: *Journal of molecular biology* 288.5
3849 (1999), pp. 911–940.
- 3850 [108] David H Mathews, Jeffrey Sabina, Michael Zuker, and
3851 Douglas H Turner. "Expanded sequence dependence of
3852 thermodynamic parameters improves prediction of RNA
3853 secondary structure." In: *Journal of Molecular Biology* 288.5
3854 (1999), pp. 911–940.
- 3855 [109] DH Matthews, TC Andre, J Kim, DH Turner, and M Zuker.
3856 "An updated recursive algorithm for RNA secondary
3857 structure prediction with improved thermodynamic pa-
3858 rameters." In: (1998).
- 3859 [110] Marco C Matthies, Stefan Bienert, and Andrew E Torda.
3860 "Dynamics in sequence space for RNA secondary struc-
3861 ture design." In: *Journal of chemical theory and computation*
3862 8.10 (2012), pp. 3663–3670.
- 3863 [111] John S McCaskill. "The equilibrium partition function
3864 and base pair binding probabilities for RNA secondary
3865 structure." In: *Biopolymers: Original Research on Biomolecules*
3866 29.6-7 (1990), pp. 1105–1119.
- 3867 [112] Nono SC Merleau and Matteo Smerlak. "A simple evo-
3868 lutionary algorithm guided by local mutations for an
3869 efficient RNA design." In: *Proceedings of the Genetic and
3870 Evolutionary Computation Conference*. 2021, pp. 1027–1034.

- 3871 [113] Nono SC Merleau and Matteo Smerlak. "An evolutionary
 3872 algorithm for inverse RNA folding inspired by Lévy flights." In: *bioRxiv* (2022).
- 3873
- 3874 [114] Gerard Minuesa, Cristina Alsina, Juan Antonio Garcia-Martin, Juan Carlos Oliveros, and Ivan Dotu. "MoiRNAiFold: a novel tool for complex in silico RNA design." In: *Nucleic acids research* 49.9 (2021), pp. 4934–4943.
- 3875
- 3876
- 3877
- 3878 [115] Peter B Moore and Thomas A Steitz. "The roles of RNA in the synthesis of protein." In: *Cold Spring Harbor perspectives in biology* 3.11 (2011), a003780.
- 3879
- 3880
- 3881 [116] Steffen Mueller, J Robert Coleman, Dimitris Papamichail, Charles B Ward, Anjaruwee Nimnual, Bruce Futcher, Steven Skiena, and Eckard Wimmer. "Live attenuated influenza virus vaccines by computer-aided rational design." In: *Nature biotechnology* 28.7 (2010), pp. 723–726.
- 3882
- 3883
- 3884
- 3885
- 3886 [117] JHA Nagel, C Flamm, IL Hofacker, K Franke, MH De Smit, P Schuster, and CWA Pleij. "Structural parameters affecting the kinetics of RNA hairpin formation." In: *Nucleic acids research* 34.12 (2006), pp. 3568–3576.
- 3887
- 3888
- 3889
- 3890 [118] Mark EJ Newman. "Power laws, Pareto distributions and Zipf's law." In: *Contemporary physics* 46.5 (2005), pp. 323–351.
- 3891
- 3892
- 3893 [119] Ruth Nussinov and Ann B Jacobson. "Fast algorithm for predicting the secondary structure of single-stranded RNA." In: *Proceedings of the National Academy of Sciences* 77.11 (1980), pp. 6309–6313.
- 3894
- 3895
- 3896
- 3897 [120] Vaitea Opuu, Nono SC Merleau, and Matteo Smerlak. "RAFFT: Efficient prediction of RNA folding pathways using the fast Fourier transform." In: *bioRxiv* (2021).
- 3898
- 3899
- 3900 [121] Jie Pan, D. Thirumalai, and Sarah A. Woodson. "Folding of RNA involves parallel pathways." In: *Journal of Molecular Biology* 273.1 (1997), pp. 7–13. doi: [10.1006/jmbi.1997.1311](https://doi.org/10.1006/jmbi.1997.1311). URL: <https://doi.org/10.1006/jmbi.1997.1311>.
- 3901
- 3902
- 3903
- 3904
- 3905 [122] Marc Parisien and Francois Major. "The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data." In: *Nature* 452.7183 (2008), pp. 51–55.
- 3906
- 3907
- 3908 [123] Fernando Portela. "An unexpectedly effective Monte Carlo technique for the RNA inverse folding problem." 2018.
- 3909

- 3910 [124] "RNACentral: a comprehensive database of non-coding
 3911 RNA sequences." In: *Nucleic acids research* 45.D1 (2017),
 3912 pp. D128–D134.
- 3913 [125] Effirul I Ramlan and Klaus-Peter Zauner. "Design of in-
 3914 teracting multi-stable nucleic acids for molecular infor-
 3915 mation processing." In: *Biosystems* 105.1 (2011), pp. 14–
 3916 24.
- 3917 [126] Jens Reeder, Peter Steffen, and Robert Giegerich. "pknot-
 3918 sRG: RNA pseudoknot folding including near-optimal
 3919 structures and sliding windows." In: *Nucleic acids research*
 3920 35.suppl_2 (2007), W320–W324.
- 3921 [127] Christian Reidys, Peter F Stadler, and Peter Schuster. "Generic
 3922 properties of combinatory maps: neutral networks of
 3923 RNA secondary structures." In: *Bulletin of mathematical
 3924 biology* 59.2 (1997), pp. 339–397.
- 3925 [128] Vladimir Reinharz, Yann Ponty, and Jérôme Waldspühl.
 3926 "A weighted sampling algorithm for the design of RNA
 3927 sequences with targeted secondary structure and nu-
 3928 cleotide distribution." In: *Bioinformatics* 29.13 (2013), pp. i308–
 3929 i315. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/btt217](https://doi.org/10.1093/bioinformatics/btt217). eprint: <https://academic.oup.com/bioinformatics/article-pdf/29/13/i308/18534655/btt217.pdf>. URL:
 3930 <https://doi.org/10.1093/bioinformatics/btt217>.
- 3931 [129] Jihong Ren, Baharak Rastegari, Anne Condon, and Hol-
 3932 ger H Hoos. "HotKnots: heuristic prediction of RNA sec-
 3933 ondary structures including pseudoknots." In: *RNA* 11.10
 3934 (2005), pp. 1494–1504.
- 3935 [130] Jessica S Reuter and David H Mathews. "RNAsstructure:
 3936 software for RNA secondary structure prediction and
 3937 analysis." In: *BMC bioinformatics* 11.1 (2010), pp. 1–9.
- 3938 [131] Andy M Reynolds. "Current status and future directions
 3939 of Lévy walk research." In: *Biology open* 7.1 (2018), bio030106.
- 3940 [132] Elena Rivas and Sean R. Eddy. "A dynamic pro-
 3941 gramming algorithm for RNA structure prediction includ-
 3942 ing pseudoknots." In: *Journal of Molecular Biology* 285.5
 3943 (1999), pp. 2053–2068. ISSN: 0022-2836. DOI: <https://doi.org/10.1006/jmbi.1998.2436>. URL: <https://www.sciencedirect.com/science/article/pii/S0022283698924366>.

- 3948 [133] Elena Rivas, Raymond Lang, and Sean R Eddy. "A range
 3949 of complex probabilistic models for RNA secondary struc-
 3950 ture prediction that includes the nearest-neighbor model
 3951 and more." In: *RNA* 18.2 (2012), pp. 193–212.
- 3952 [134] Debra L Robertson and Gerald F Joyce. "Selection in
 3953 vitro of an RNA enzyme that specifically cleaves single-
 3954 stranded DNA." In: *Nature* 344.6265 (1990), pp. 467–468.
- 3955 [135] John M Rosenberg, Nadrian C Seeman, Roberta O Day,
 3956 and Alexander Rich. "RNA double-helical fragments at
 3957 atomic resolution: II. The crystal structure of sodium
 3958 guanylyl-3', 5'-cytidine nonahydrate." In: *Journal of molec-*
 3959 *ular biology* 104.1 (1976), pp. 145–167.
- 3960 [136] Frederic Runge, Danny Stoll, Stefan Falkner, and Frank
 3961 Hutter. "Learning to design RNA." In: *arXiv preprint arXiv:1812.11951*
 3962 (2018).
- 3963 [137] Rick Russell, Xiaowei Zhuang, Hazen P Babcock, Ian S
 3964 Millett, Sebastian Doniach, Steven Chu, and Daniel Her-
 3965 schlag. "Exploring the folding landscape of a structured
 3966 RNA." In: *Proceedings of the National Academy of Sciences*
 3967 99.1 (2002), pp. 155–160.
- 3968 [138] Yasubumi Sakakibara, Michael Brown, Richard Hughey,
 3969 I Saira Mian, Kimmen Sjölander, Rebecca C Underwood,
 3970 and David Haussler. "Stochastic context-free grammars
 3971 for tRNA modeling." In: *Nucleic acids research* 22.23 (1994),
 3972 pp. 5112–5120.
- 3973 [139] Tore Samuelsson and Christian Zwieb. "The signal recog-
 3974 nition particle database (SRPDB)." In: *Nucleic Acids Re-*
 3975 *search* 27.1 (1999), pp. 169–170.
- 3976 [140] Baby Santosh, Akhil Varshney, and Pramod Kumar Ya-
 3977 dava. "Non-coding RNAs: biological functions and ap-
 3978 plications." In: *Cell biochemistry and function* 33.1 (2015),
 3979 pp. 14–22.
- 3980 [141] Kengo Sato, Manato Akiyama, and Yasubumi Sakakibara.
 3981 "RNA secondary structure prediction using deep learning
 3982 with thermodynamic integration." In: *Nature communica-*
 3983 *tions* 12.1 (2021), pp. 1–9.

- 3984 [142] Kengo Sato, Yuki Kato, Michiaki Hamada, Tatsuya Akutsu,
 3985 and Kiyoshi Asai. "IPknot: fast and accurate prediction
 3986 of RNA secondary structures with pseudoknots using
 3987 integer programming." In: *Bioinformatics* 27.13 (2011),
 3988 pp. i85–i93.
- 3989 [143] Martin Sauvageau, Loyal A Goff, Simona Lodato, Boyan
 3990 Bonev, Abigail F Groff, Chiara Gerhardinger, Diana B
 3991 Sanchez-Gomez, Ezgi Hacisuleyman, Eric Li, Matthew
 3992 Spence, et al. "Multiple knockout mouse models reveal
 3993 lincRNAs are required for life and brain development."
 3994 In: *elife* 2 (2013), e01749.
- 3995 [144] Murray N Schnare, Simon H Damberger, Michael W Gray,
 3996 and Robin R Gutell. "Comprehensive comparison of struc-
 3997 tural characteristics in eukaryotic cytoplasmic large sub-
 3998 unit (23 S-like) ribosomal RNA." In: *Journal of Molecular*
 3999 *Biology* 256.4 (1996), pp. 701–719.
- 4000 [145] Peter Schuster, Walter Fontana, Peter F Stadler, and Ivo L
 4001 Hofacker. "From sequences to shapes and back: a case
 4002 study in RNA secondary structures." In: *Proceedings of the*
 4003 *Royal Society of London. Series B: Biological Sciences* 255.1344
 4004 (1994), pp. 279–284.
- 4005 [146] Nadrian C Seeman, John M Rosenberg, FL Suddath, Jung
 4006 Ja Park Kim, and Alexander Rich. "RNA double-helical
 4007 fragments at atomic resolution: I. The crystal and molec-
 4008 ular structure of sodium adenylyl-3', 5'-uridine hexahy-
 4009 drate." In: *Journal of molecular biology* 104.1 (1976), pp. 109–
 4010 144.
- 4011 [147] Matthew G Seetin and David H Mathews. "RNA struc-
 4012 ture prediction: an overview of methods." In: *Bacterial*
 4013 *Regulatory RNA* (2012), pp. 99–122.
- 4014 [148] Martin J Serra and Douglas H Turner. "Predicting thermo-
 4015 dynamic properties of RNA." In: *Methods in enzymology*.
 4016 Vol. 259. Elsevier, 1995, pp. 242–261.
- 4017 [149] Bruce A Shapiro and Kaizhong Zhang. "Comparing mul-
 4018 tiple RNA secondary structures using tree comparisons."
 4019 In: *Bioinformatics* 6.4 (1990), pp. 309–318.
- 4020 [150] Vishnu Prakash Sharma, Harji Ram Choudhary, Sandeep
 4021 Kumar, and Vikas Choudhary. "A modified DE: Popula-
 4022 tion or generation based levy flight differential evolution
 4023 (PGLFDE)." In: *2015 International Conference on Futuristic*

- 4024 *Trends on Computational Analysis and Knowledge Manage-*
 4025 *ment (ABLAZE)*. IEEE. 2015, pp. 704–710.
- 4026 [151] Jade Shi, Rhiju Das, and Vijay S Pande. “SentRNA: Im-
 4027 proving computational RNA design by incorporating a
 4028 prior of human design strategies.” 2018.
- 4029 [152] Micheal F Shlesinger, George M Zaslavsky, and Uriel
 4030 Frisch. *Lévy flights and related topics in physics*. Vol. 450.
 4031 Berlin, Heidelberg: Springer Berlin Heidelberg, 1995.
- 4032 [153] Wenjie Shu, Ming Liu, Hebing Chen, Xiaochen Bo, and
 4033 Shengqi Wang. “ARDesigner: a web-based system for
 4034 allosteric RNA design.” In: *Journal of biotechnology* 150.4
 4035 (2010), pp. 466–473.
- 4036 [154] Christian Höner zu Siederdissen, Stephan H Bernhart, Pe-
 4037 ter F Stadler, and Ivo L Hofacker. “A folding algorithm for
 4038 extended RNA secondary structures.” In: *Bioinformatics*
 4039 27.13 (2011), pp. i129–i136.
- 4040 [155] Michael F Sloma and David H Mathews. “Exact calcula-
 4041 tion of loop formation probability identifies folding mo-
 4042 tifs in RNA secondary structures.” In: *RNA* 22.12 (2016),
 4043 pp. 1808–1818.
- 4044 [156] Michael F Sloma and David H Mathews. “Base pair prob-
 4045 ability estimates improve the prediction accuracy of RNA
 4046 non-canonical base pairs.” In: *PLoS computational biology*
 4047 13.11 (2017), e1005827.
- 4048 [157] Sergey V. Solomatin, Max Greenfeld, Steven Chu, and
 4049 Daniel Herschlag. “Multiple native states reveal persis-
 4050 tent ruggedness of an RNA folding landscape.” In: *Nature*
 4051 463.7281 (2010), pp. 681–684. DOI: [10.1038/nature08717](https://doi.org/10.1038/nature08717).
 4052 URL: <https://doi.org/10.1038/nature08717>.
- 4053 [158] T. Specht, M. Szymanski, M. Z. Barciszewska, J. Barciszewski,
 4054 and V. A. Erdmann. “Compilation of 5s rRNA and 5s
 4055 rRNA gene sequences.” In: *Nucleic Acids Research* 25.1
 4056 (1997), pp. 96–97. DOI: [10.1093/nar/25.1.96](https://doi.org/10.1093/nar/25.1.96). URL: <https://doi.org/10.1093/nar/25.1.96>.
- 4058 [159] Mathias Sprinzl, Carsten Horn, Melissa Brown, Anatoli
 4059 Ioudovitch, and Sergey Steinberg. “Compilation of tRNA
 4060 sequences and sequences of tRNA genes.” In: *Nucleic
 4061 Acids Research* 26.1 (1998), pp. 148–153.

- 4062 [160] Evan W Steeg. "Neural networks, adaptive optimization,
 4063 and RNA secondary structure prediction." In: *Artificial
 4064 intelligence and molecular biology* (1993), pp. 121–60.
- 4065 [161] Paul R Stein and Michael S Waterman. "On some new se-
 4066 quences generalizing the Catalan and Motzkin numbers." In: *Discrete Mathematics* 26.3 (1979), pp. 261–272.
- 4068 [162] Sergei Svitashev, Joshua K Young, Christine Schwartz,
 4069 Huirong Gao, S Carl Falco, and A Mark Cigan. "Targeted
 4070 mutagenesis, precise gene editing, and site-specific gene
 4071 insertion in maize using Cas9 and guide RNA." In: *Plant
 4072 physiology* 169.2 (2015), pp. 931–945.
- 4073 [163] Yoshiyasu Takefuji and L Chen. "Parallel algorithms for
 4074 finding a near-maximum independent set of." In: *IEEE
 4075 Trans. Neural Networks* 1.3 (1990), p. 263.
- 4076 [164] Akito Taneda. "MODENA: a multi-objective RNA inverse
 4077 folding." In: *Advances and applications in bioinformatics and
 4078 chemistry: AABC* 4 (2011), p. 1.
- 4079 [165] Akito Taneda. "Multi-Objective Genetic Genetic for Pseu-
 4080 doknotted RNA Sequence Design." In: *Frontiers in Ge-
 4081 netics* 3 (2012), p. 36. ISSN: 1664-8021. doi: [10 . 3389 /
 4082 fgene . 2012 . 00036](https://doi.org/10.3389/fgene.2012.00036). URL: [https : / / www . frontiersin .
 4083 org / article / 10 . 3389 / fgene . 2012 . 000INFO - RNA36](https://www.frontiersin.org/article/10.3389/fgene.2012.000INFO-RNA36).
- 4084 [166] Akito Taneda. "Multi-objective optimization for RNA de-
 4085 sign with multiple target secondary structures." In: *BMC
 4086 bioinformatics* 16.1 (2015), pp. 1–20.
- 4087 [167] Michela Taufer, Abel Licon, Roberto Araiza, David Mire-
 4088 les, FHD Van Batenburg, Alexander P Gulyaev, and
 4089 Ming-Ying Leung. "PseudoBase++: an extension of Pseu-
 4090 doBase for easy searching, formatting and visualization
 4091 of pseudoknots." In: *Nucleic acids research* 37.suppl_1 (2009),
 4092 pp. D127–D135.
- 4093 [168] Siqi Tian and Rhiju Das. "RNA structure through multi-
 4094 dimensional chemical mapping." In: *Quarterly reviews of
 4095 biophysics* 49 (2016).
- 4096 [169] Pilar Tijerina, Sabine Mohr, and Rick Russell. "DMS foot-
 4097 printing of structured RNAs and RNA–protein complexes."
 4098 In: *Nature protocols* 2.10 (2007), pp. 2608–2623.
- 4099 [170] Ignacio Tinoco Jr and Carlos Bustamante. "How RNA
 4100 folds." In: *Journal of molecular biology* 293.2 (1999), pp. 271–
 4101 281.

- 4102 [171] Ignacio Tinoco, Olke C. Uhlenbeck, and Mark D. Levine.
 4103 "Estimation of Secondary Structure in Ribonucleic Acids."
 4104 In: *Nature* 230.5293 (Apr. 1971). Number: 5293 Publisher:
 4105 Nature Publishing Group, pp. 362–367. ISSN: 1476-4687.
 4106 DOI: [10.1038/230362a0](https://doi.org/10.1038/230362a0). URL: <https://www.nature.com/articles/230362a0> (visited on 04/14/2021).
- 4108 [172] Craig Tuerk and Larry Gold. "Systematic evolution of
 4109 ligands by exponential enrichment: RNA ligands to bac-
 4110 teriophage T4 DNA polymerase." In: *science* 249.4968
 4111 (1990), pp. 505–510.
- 4112 [173] Douglas H. Turner and David H. Mathews. "NNDB: the
 4113 nearest neighbor parameter database for predicting sta-
 4114 bility of nucleic acid secondary structure." In: *Nucleic
 4115 Acids Research* 38.suppl_1 (2009), pp. D280–D282.
- 4116 [174] Douglas H Turner and David H Mathews. "NNDB: the
 4117 nearest neighbor parameter database for predicting sta-
 4118 bility of nucleic acid secondary structure." In: *Nucleic
 4119 Acids Research* 38.suppl_1 (2010), pp. D280–D282.
- 4120 [175] Sinan Uğur Umu and Paul P Gardner. "A comprehensive
 4121 benchmark of RNA–RNA interaction prediction tools for
 4122 all domains of life." In: *Bioinformatics* 33.7 (2017), pp. 988–
 4123 996.
- 4124 [176] Jason G Underwood, Andrew V Uzilov, Sol Katzman,
 4125 Courtney S Onodera, Jacob E Mainzer, David H Math-
 4126 ewes, Todd M Lowe, Sofie R Salama, and David Haussler.
 4127 "FragSeq: transcriptome-wide RNA structure probing us-
 4128 ing high-throughput sequencing." In: *Nature methods* 7.12
 4129 (2010), pp. 995–1001.
- 4130 [177] Gandhimohan M Viswanathan, EP Raposo, and MGE Da
 4131 Luz. "Lévy flights and superdiffusion in the context of
 4132 biological encounters and random searches." In: *Physics
 4133 of Life Reviews* 5.3 (2008), pp. 133–150.
- 4134 [178] Alexey G Vitreschak, Dmitry A Rodionov, Andrey A
 4135 Mironov, and Mikhail S Gelfand. "Riboswitches: the old-
 4136 est mechanism for the regulation of gene expression?"
 4137 In: *Trends in Genetics* 20.1 (2004), pp. 44–50.
- 4138 [179] Manja Wachsmuth, Gesine Domin, Ronny Lorenz, Robert
 4139 Serfling, Sven Findeiß, Peter F Stadler, and Mario Mörl.
 4140 "Design criteria for synthetic riboswitches acting on tran-
 4141 scription." In: *RNA biology* 12.2 (2015), pp. 221–231.

- [180] Haoyi Wang, Hui Yang, Chikdu S Shivalila, Meelad M Dawlaty, Albert W Cheng, Feng Zhang, and Rudolf Jaenisch. "One-step generation of mice carrying mutations in multiple genes by CRISPR/Cas-mediated genome engineering." In: *cell* 153.4 (2013), pp. 910–918.
- [181] Shouhua Wang, Ting ting Su, Huanjun Tong, Weibin Shi, Fei Ma, and Zhiwei Quan. "CircPVT1 promotes gallbladder cancer growth by sponging miR-339-3p and regulates MCL-1 expression." In: *Cell Death Discovery* 7.1 (2021), pp. 1–10.
- [182] Zhong Wang, Mark Gerstein, and Michael Snyder. "RNA-Seq: a revolutionary tool for transcriptomics." In: *Nature reviews genetics* 10.1 (2009), pp. 57–63.
- [183] Richard B Waring and R Wayne Davies. "Assessment of a model for intron RNA secondary structure relevant to RNA self-splicing—a review." In: *Gene* 28.3 (1984), pp. 277–291.
- [184] James D Watson and Francis HC Crick. "Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid." In: *Nature* 171.4356 (1953), pp. 737–738.
- [185] Lina Weinbrand, Assaf Avihoo, and Danny Barash. "RNAbinv: an interactive Java application for fragment-based design of RNA sequences." In: *Bioinformatics* 29.22 (2013), pp. 2938–2940.
- [186] Eric Westhof and Valérie Fritsch. "RNA folding: beyond Watson–Crick pairs." In: *Structure* 8.3 (2000), R55–R65. ISSN: 0969-2126. DOI: [https://doi.org/10.1016/S0969-2126\(00\)00112-X](https://doi.org/10.1016/S0969-2126(00)00112-X). URL: <https://www.sciencedirect.com/science/article/pii/S096921260000112X>.
- [187] Kevin A Wilkinson, Edward J Merino, and Kevin M Weeks. "Selective 2'-hydroxyl acylation analyzed by primer extension (SHAPE): quantitative RNA structure analysis at single nucleotide resolution." In: *Nature protocols* 1.3 (2006), pp. 1610–1616.
- [188] Wade C Winkler and Ronald R Breaker. "Genetic control by metabolite-binding riboswitches." In: *Chembiochem* 4.10 (2003), pp. 1024–1032.
- [189] SA Woodson. "Recent insights on RNA folding mechanisms from catalytic RNA." In: *Cellular and Molecular Life Sciences CMLS* 57.5 (2000), pp. 796–808.

- 4182 [190] Xiufeng Yang, Kazuki Yoshizoe, Akito Taneda, and Koji
 4183 Tsuda. "RNA inverse folding using Monte Carlo tree
 4184 search." In: *BMC bioinformatics* 18.1 (2017), p. 468.
- 4185 [191] Hua-Ting Yao, Cedric Chauve, Mireille Regnier, and Yann
 4186 Ponty. "Exponentially few RNA structures are designable."
 4187 In: *Proceedings of the 10th ACM International Conference on*
 4188 *Bioinformatics, Computational Biology and Health Informatics*.
 4189 2019, pp. 289–298.
- 4190 [192] Hua-Ting Yao, Jérôme Waldspühl, Yann Ponty, and Se-
 4191 bastian Will. "Taming Disruptive Base Pairs to Reconcile
 4192 Positive and Negative Structural Design of RNA." In:
 4193 *RECOMB 2021-25th international conference on research in*
 4194 *computational molecular biology*. 2021.
- 4195 [193] Joseph N Zadeh, Conrad D Steenberg, Justin S Bois, Brian
 4196 R Wolfe, Marshall B Pierce, Asif R Khan, Robert M Dirks,
 4197 and Niles A Pierce. "NUPACK: Analysis and design of
 4198 nucleic acid systems." In: *Journal of computational chemistry*
 4199 32.1 (2011), pp. 170–173.
- 4200 [194] Joseph N Zadeh, Brian R Wolfe, and Niles A Pierce. "Nu-
 4201 cleic acid sequence design via efficient ensemble defect
 4202 optimization." In: *Journal of computational chemistry* 32.3
 4203 (2011), pp. 439–452.
- 4204 [195] Shay Zakov, Yoav Goldberg, Michael Elhadad, and Michal
 4205 Ziv-Ukelson. "Rich parameterization improves RNA struc-
 4206 ture prediction." In: *Journal of Computational Biology* 18.11
 4207 (2011), pp. 1525–1542.
- 4208 [196] Wenbing Zhang and Shi-Jie Chen. "RNA hairpin-folding
 4209 kinetics." In: *Proceedings of the National Academy of Sciences*
 4210 99.4 (2002), pp. 1931–1936.
- 4211 [197] Wenbing Zhang and Shi-Jie Chen. "Analyzing the biopoly-
 4212 mer folding rates and pathways using kinetic cluster
 4213 method." In: *The Journal of chemical physics* 119.16 (2003),
 4214 pp. 8716–8729.
- 4215 [198] Wenbing Zhang and Shi-Jie Chen. "Exploring the com-
 4216 plex folding kinetics of RNA hairpins: I. General fold-
 4217 ing kinetics analysis." In: *Biophysical journal* 90.3 (2006),
 4218 pp. 765–777.

- 4219 [199] Wenbing Zhang and Shi-Jie Chen. "Exploring the com-
4220 plex folding kinetics of RNA hairpins: I. General fold-
4221 ing kinetics analysis." In: *Biophysical Journal* 90.3 (2006),
4222 pp. 765–777.
- 4223 [200] Qi Zhao, Zheng Zhao, Xiaoya Fan, Zhengwei Yuan, Qian
4224 Mao, and Yudong Yao. "Review of machine learning
4225 methods for RNA secondary structure prediction." In:
4226 *PLoS computational biology* 17.8 (2021), e1009291.
- 4227 [201] Yu Zhu, ZhaoYang Xie, YiZhou Li, Min Zhu, and Yi-Ping
4228 Phoebe Chen. "Research on folding diversity in statistical
4229 learning methods for RNA secondary structure predic-
4230 tion." In: *International Journal of Biological Sciences* 14.8
4231 (2018), p. 872.
- 4232 [202] Michael Zuker and David Sankoff. "RNA secondary struc-
4233 tures and their prediction." In: *Bulletin of mathematical
4234 biology* 46.4 (1984), pp. 591–621.
- 4235 [203] Michael Zuker and Patrick Stiegler. "Optimal computer
4236 folding of large RNA sequences using thermodynamics
4237 and auxiliary information." In: *Nucleic acids research* 9.1
4238 (1981), pp. 133–148.
- 4239 [204] C. Zwieb. "Tmrdb (tmRNA database)." In: *Nucleic Acids
4240 Research* 28.1 (2000), pp. 169–170. doi: [10.1093/nar/28.1.169](https://doi.org/10.1093/nar/28.1.169). URL: <https://doi.org/10.1093/nar/28.1.169>.
- 4242 [205] C. Zwieb. "Tmrdb (tmRNA database)." In: *Nucleic Acids
4243 Research* 31.1 (2003), pp. 446–447. doi: [10.1093/nar/gkg019](https://doi.org/10.1093/nar/gkg019). URL: <https://doi.org/10.1093/nar/gkg019>.

⁴²⁴⁵ DECLARATION

⁴²⁴⁶ Put your declaration here.

⁴²⁴⁷ *Leipzig, June 2022*

⁴²⁴⁸

Nono Saha Cyrille Merleau

[June 17, 2022 at 17:55 – 1.0]

4249

4250 **COLOPHON**

4251 This document was typeset using the typographical look-and-
4252 feel **classicthesis** developed by André Miede and Ivo Pletikosić.
4253 The style was inspired by Robert Bringhurst's seminal book on
4254 typography "*The Elements of Typographic Style*". **classicthesis** is
4255 available for both L^AT_EX and LyX:

4256 <https://bitbucket.org/amiede/classicthesis/>

4257 Happy users of **classicthesis** usually send a real postcard to
4258 the author, a collection of postcards received so far is featured
4259 here:

4260 <http://postcards.miede.de/>

4261 Thank you very much for your feedback and contribution.