HAL open science

Local decomposition in RNA structural design

Hua-Ting Yao

To cite this version:

Hua-Ting Yao. Local decomposition in RNA structural design. Bioinformatics [q-bio.QM]. Institut Polytechnique de Paris; McGill university (Montréal, Canada), 2021. English. NNT : 2021IPPAX126 . tel-03538576v3

HAL Id: tel-03538576

https://theses.hal.science/tel-03538576v3

Submitted on 12 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAX126

Thèse de doctorat

Local Decomposition In RNA Structural Design

Thèse de doctorat de l'Institut Polytechnique de Paris et de l'université McGill
préparée à l'École polytechnique

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Palaiseau, le 15/12/2021, par

HUA-TING YAO

Composition du Jury :

Sebastian Will Professeur, LIX, École Polytechnique	Président
Christine Heitsch Professeur, School of Mathematics, Georgia Institute of Technology	Rapporteur
Aïda Ouangraoua Professeur, Department of Computer Science, Université de Sherbrooke	Rapporteur
Peter Stadler Professeur, Faculty of Mathematics and Computer Science, Université de Leipzig	Examinateur
Yann Ponty Directeur de Recherche, CNRS-LIX, École Polytechnique	Directeur de thèse
Jérôme Waldispühl Professeur, School of Computer Science, Université McGill	Co-directeur de thèse
Mireille Régnier Directrice, INRIA, Lille	Invité

LOCAL DECOMPOSITION IN RNA STRUCTURAL DESIGN

HUA-TING YAO

LIX – Laboratoire d'Informatique
École Polytechnique, Palaiseau, France

School of Computer Science
McGill University, Montreal, Canada

Supervisors:

Yann Ponty, Directeur de Recherche, CNRS
Jérôme Waldispühl, Associate Professor, McGill University

October 2021

A thesis submitted to Ecole Polytechnique and McGill University in partial fulfillment of the requirements of the degree of Doctor of Philosophy

© Hua-Ting Yao 2021

Hua-Ting Yao: *Local Decomposition in RNA Structural Design*

Ohana means family.
Family means nobody gets left behind, or forgotten.

— Lilo & Stitch

Dedicated to the loving memory of my mom, 권주임.

ABSTRACT

RNA positive structural design problem attempts to find RNA sequences achieving low free energy of the target secondary structure. Differently, in the negative design, solution sequences should adopt the target structure as its folding preferentially to any alternative structure, according to the given metric and energy model. Inverse folding, a typical negative design, requires the target to be the solution sequence's Minimal Free Energy (MFE) folding. Other metrics, like the ensemble defect, are also considered for design evaluation.

The additivity of the energy model suggests the existence of local properties for the RNA design problem. It was discovered in several works that, due to the presence of specific local motifs, some secondary structures are undesignable, *i.e.*, no RNA sequence can fold into the target structure while satisfying the negative design objective. The sequence sampling approach is often used in the positive design. Unwanted local structures, like base pairs, repeatedly form while folding sampled sequences toward the negative design. In this thesis, we study the impact of such local nature on the combinatorial aspect and on the development of negative design methods.

We show that the proportion of designable secondary structures decreases exponentially with the target structure length from the combinatorial aspect. Given a negative design metric, we propose an automated pipeline to identify all undesignable motifs. Enumerating secondary structures avoiding such local obstructions followed by asymptotic analysis yields an upper-bounds on the number of designable structures. In addition, we define a lower bound for the structural ensemble defect derived from occurred local motifs. We show that the lower bound follows a Normal limiting distribution with a closed-form expression, implying also an exponential decrease.

We then present InfraRed, a generic framework for efficient combinatorial sampling. We formalize the RNA design problem as a Constraint Satisfaction Problem (CSP) with design objectives described as a set of constraints and a set of weighted functions. Assignments satisfying constraints are generated from a Boltzmann weighted distribution using a dynamic programming algorithm followed by stochastic backtracking. The approach is Fixed-Parameter Trackable (FPT) for the treewidth of the dependency graph induced from the problem. We show that the framework can be easily employed for RNA positive design and flexible applications.

Finally, as an application of Infrared, we propose an original iterative sampling approach that captures negative design principles implemented in RNA POsitive and Negative Design (RNAPOND). A set of Disruptive Base Pairs (DBPs) is identified at each round and subsequently prevented from pairing by introducing proper constraints into the sampling framework. Despite the NP-hardness of the associated decision

problem, an efficient sequence sampling algorithm is ensured by the Infrared framework. Our approach achieves a similar or better success rate than state-of-the-art negative design tools while allowing for the generation of diverse, thermodynamically efficient designs, *i.e.*, positive design principles.

One of the research directions of the works presented in this thesis is the extension to more complicated structures, such as pseudoknotted secondary structures. The flexibility of the InfraRed framework opens a door for design tool development. For example, the success of RNAPOND suggests a potential approach for RNA negative structural design.

RÉSUMÉ SUBSTANTIEL

Le problème de design structural positif de l'ARN tente de trouver des séquences d'ARN réalisant une faible énergie libre de la structure secondaire cible. Par contre, dans le problème de design négatif, les séquences de solution doivent adopter la structure cible comme repliement préférentiellement à toute structure alternative. Le problème du repliement d'inverse, un problème typique de design négatif, exige que la cible soit la structure secondaire ayant l'énergie libre minimale (MFE) de la solution. D'autres métriques, telles que le défaut d'ensemble, sont également prises en compte pour l'évaluation de la séquence réalisée.

L'additivité du modèle d'énergie suggère l'existence de propriétés locales pour le problème de design de l'ARN. Il a été découvert dans plusieurs travaux que, en raison de la présence de certains motifs locaux, aucune séquence d'ARN ne peut se replier dans la structure cible tout en satisfaisant l'objectif de design négatif. L'approche d'échantillonnage de séquence est souvent utilisée dans le design positif. Les structures locales irréalisables, comme les paires de bases, se forment de manière répétée lors du repliement des séquences échantillonnées en considérant le design négatif. Dans cette thèse, nous étudions l'impact de cette nature locale sur l'aspect combinatoire et sur le développement de méthodes de design négatif.

Nous montrons que la proportion de structures secondaires réalisables diminue de façon exponentiellement avec la longueur de la structure cible du point de vue combinatoire. Étant donné une métrique de design négatif, nous proposons un schéma automatisé pour identifier tous les motifs non réalisables. L'énumération des structures secondaires évitant ces obstructions locales, suivie d'une analyse asymptotique, permet d'obtenir une borne supérieure du nombre de structures réalisables. En outre, nous définissons une borne inférieure pour le défaut d'ensemble structural dérivé des motifs locaux apparus. Nous montrons que cette borne inférieure suit une distribution limite Gaussienne avec une expression explicite, ce qui implique aussi la diminution exponentielle.

Nous présentons ensuite InfraRed, un système générique d'échantillonnage combinatoire efficace. Nous formalisons le problème de design de l'ARN comme un problème de CSP avec des objectifs de design décrits comme un ensemble de contraintes et un ensemble de fonctions pondérées. Les évaluations des variables satisfaisant les contraintes sont générées à partir d'une distribution pondérée de Boltzmann en utilisant un algorithme de programmation dynamique suivi d'un backtrack stochastique. L'approche est en classe de FPT pour la largeur arborescente du graphe de dépendance induit par le problème. Nous montrons que ce cadre peut être facilement employé pour le design positif de l'ARN et les applications variées.

Enfin, en tant qu'application du système InfraRed, nous proposons une approche originale d'échantillonnage itératif qui capture les principes de design négatif mis en

œuvre dans RNAPOND. Un ensemble de paires de bases perturbatrices est identifié à chaque tour et on les empêche ensuite de s'apparier en introduisant des contraintes appropriées dans le cadre de l'échantillonnage. Malgré que le problème de décision associé est NP-difficile, un algorithme d'échantillonnage de séquence efficace est garanti par le système InfraRed. Notre approche atteint un taux de réussite similaire ou supérieur aux états de l'art, tout en permettant la génération de séquences diverses et thermodynamiquement efficaces, c'est-à-dire des principes de design positif.

L'un des axes de recherche des travaux présentés dans cette thèse est l'extension à des structures plus complexes, telles que les structures secondaires contenant pseudonœuds. La flexibilité du système InfraRed ouvre une porte au développement d'outils de design. Par exemple, le succès de RNAPOND suggère une approche potentielle pour la design structural négatif d'ARN.

PUBLICATIONS

Some elements of this thesis have been published in the following venues:

- A conference article for ACM-BCB'19 (Chapter 5 and 6)

  Hua-Ting Yao, Cedric Chauve, Mireille Regnier, and Yann Ponty. "Exponentially few RNA structures are designable." In: *ACM-BCB 2019 - 10th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. Niagara-Falls, United States: ACM Press, Sept. 2019, pp. 289–298. DOI: 10.1145/3307339.3342163. URL: https://hal.inria.fr/hal-02141853

- A book chapter illustrating the usage of RNARedPrint (Chapter 8)

  Yann Ponty, Stefan Hammer, Hua-Ting Yao, and Sebastian Will. "Advanced design of structural RNAs using RNARedPrint." In: *RNA Bioinformatics*. Ed. by Ernesto Picardi. Methods in Molecular Biology. 2020. URL: https://hal.inria.fr/hal-02990264

- A conference article for RECOMB'21 (Chapter 9)

  Hua-Ting Yao, Jérôme Waldispühl, Yann Ponty, and Sebastian Will. "Taming Disruptive Base Pairs to Reconcile Positive and Negative Structural Design of RNA." In: *RECOMB 2021 - 25th international conference on research in computational molecular biology*. Padova, France, Apr. 2021. URL: https://hal.inria.fr/hal-02987566

In addition to these published works, I also contributed to the following work, not included in this thesis:

- A conference article for RECOMB'20 that is not included in this thesis

  Roman Sarrazin-Gendron, Hua-Ting Yao, Vladimir Reinharz, Carlos G Oliver, Yann Ponty, and Jérôme Waldispühl. "Stochastic Sampling of Structural Contexts Improves the Scalability and Accuracy of RNA 3D Modules Identification." In: *RECOMB 2020 - 24th Annual International Conference on Research in Computational Molecular Biology*. Proceedings of RECOMB - 24th Annual International Conference on Research in Computational Molecular Biology - 2020. Padova, Italy, May 2020. URL: https://hal.inria.fr/hal-02354733

無論什麼事,得之於人者太多,出之於己者太少.
因為需要感謝的人太多了,就感謝天罷.

— 陳之藩《謝天》

ACKNOWLEDGEMENTS

The realization of a doctoral thesis is a very long and challenging process. Despite the effort and time invested, such a journey cannot be accomplished without various help.

First, I would like to offer my deepest gratitude to my Ph.D. supervisor, Yann Ponty, for his continuous support and guidance during my Ph.D. period. His enthusiasm and patience for research set a standard for me to follow. His advice allows me to develop the fundamental skills to complete this thesis and continue academic research in the future. Beyond science, he introduces me to the world of climbing. He is not only a supervisor but also a life mentor for me.

I would also like to express my sincere gratitude to Jérôme Waldispühl, my co-supervisor, for his support and kindness. Without his help, it would be impossible to succeed during the period in Montreal during the first year of my Ph.D. The discussion on various subjects with him gives different suggestions and ideas for research.

I thank the rapporteurs, Christine Heitsch and Aïda Ouangraoua, for dedicating their time to reading my thesis' manuscript, writing the detailed reports, and pointing out necessary corrections despite the complicated circumstance. I also thank Peter Stadler and Mireille Régnier for being my jury members. In particular, I would like to thank Sebastian Will for being the president of the jury and for numerous discussions during my Ph.D., from whom I have learned a lot.

I want to thank the supervisors of my previous internships, Cedric Chauve, Leonid Chindelevitch, Laurent Guyon, and my high school teacher, 劉翠華, for their guidance. Thanks to them, I have learned different skills that support me to the current moment. I also want to thank the team members for all discussions, talks, and time we spent together: Sarah Berkemer, Bertrand, Juraj, and Afaf from Ecole Polytechnique; Mathieu Blanchette, Vladimir Reinharz, Roman, Jacques, Vincent, and Carlos from different labs in Montreal. Notably, I want to thank Roman for guiding me through the life of McGill University and Montreal.

For personal life, I want to thank Gwenola and Oksenberg family, Laurent, Cathérine, and Benjamin, for all the help in the first two years when I arrived in France. Thanks to all baseball clubs' teammates and coaches, especially Manu, Gauthiou, and Nicolas, from Les Gothics de Gif-sur-Yvette, Les Grizzlys de Grenoble, and Les Suricates de Clamart. I also want to thank 鈺霖, 杜力, 皆安, 治綱, 瑞婷, Bamboo, and Livi for

the time we spent on Bridge and video games during the pandemic; Benjamin, Marc, Maïwenn, Pauline, and Manon for numerous climbing times; Emily, 紘翎, 莉雯, 志煌, 嘉蔓, 小黑, 林岳, Kyo, 有蓉, Ssu-Ting Lai, Zhikai Wang, 阿光, 十五區金城武, Natalie, Maud, Robin, and Timothée for all the help and talks. In addition, special thanks to the podcast 魚子醬的《鍵盤考古指南》.

Last but not least, I would like to thank my family, especially my mom, for her support and her love and my uncles and aunts (in-laws) for taking care of me whenever I visit Korea. Finally, I want to thank 宛儒, my love, for all her supports and encouragement for the past few years. Without them, I could not have enough strength to follow my dream.

CONTENTS

1 INTRODUCTION 1

I BACKGROUND KNOWLEDGE 9

2 RNA 2D STRUCTURE PREDICTION 11
- 2.1 Notion Related to RNA Bioinformatics 11
- 2.2 Secondary Structure Prediction with Energy Minimization 13
- 2.3 Boltzmann Distribution Paradigm 17
- 2.4 Ensemble representatives and expected distance 21

3 RNA STRUCTURAL DESIGN 25
- 3.1 Design Objectives 25
- 3.2 State Of The Art 28

4 ANALYTIC COMBINATORICS 31
- 4.1 Formal Language 31
- 4.2 Analytic Combinatorics 34

II UPPER-BOUND FOR DESIGNABLE RNA 2D STRUCTURES 43

5 UNDESIGNABLE MOTIFS 45
- 5.1 Motif definition 45
- 5.2 Local obstructions 53
- 5.3 Undesignable and hard motifs in experimentally-determined 3D structures 63

6 ENUMERATING SECONDARY STRUCTURES AVOIDING LOCAL OBSTRUCTIONS 67
- 6.1 Grammar and generating function 68
- 6.2 Computing and estimating the dominant singularity 70
- 6.3 Upper-bound for designable secondary structures 72

7 BOUNDING THE ASYMPTOTIC DISTRIBUTION OF SUPERADDITIVE DEFECTS 77
- 7.1 Superadditive ensemble defect 77
- 7.2 Bivariate analysis of ensemble defect with minimum distance $\theta = 0$. 79
- 7.3 New ensemble defect lower bound with full motif set 98

III APPLICATION OF TREE DECOMPOSITIONS TO RNA DESIGN 101

8 INFRARED 103
- 8.1 Design Problem as Constraint Satisfaction Problem 103
- 8.2 InfraRed Core Engine 106
- 8.3 Implementation and Usage 113
- 8.4 Finite State Automata in InfraRed 116

9 RNA POSITIVE AND NEGATIVE DESIGN (RNAPOND) 119
- 9.1 Method 120
- 9.2 Complexity aspects 122

9.3 Validation and comparison to state of the art 126

IV CONCLUSION AND PERSPECTIVES 131

10 CONCLUSION AND PERSPECTIVES 133

10.1 Conclusion 133

10.2 Perspectives 134

BIBLIOGRAPHY 139

LIST OF FIGURES

Figure 1.1 Simplified chemical form of pseudoknotted structure 2
Figure 1.2 Base pair stack in 3D 3
Figure 1.3 Positive and negative design illustration 4
Figure 2.1 Invalid secondary structures 12
Figure 2.2 Secondary structure representations 13
Figure 2.3 Classic secondary structure decomposition 15
Figure 2.4 Secondary structure loop decompsition 16
Figure 2.5 Outside region decomposition 21
Figure 5.1 Undesignable motifs 45
Figure 5.2 Example of motif 46
Figure 5.3 Examples of motif overlapping 49
Figure 5.4 Support figure for local ensemble defect proof 51
Figure 5.5 Minimal completion and trimming operation 54
Figure 5.6 Workflow to identify undesignable motifs 57
Figure 5.7 Motif and local obstruction amount in length 60
Figure 5.8 Local obstructions with suboptimal defect 61
Figure 5.9 Histogram of suboptimal defect 61
Figure 5.10 Local obstructions with probability defect 62
Figure 5.11 Distribution of motif probability defects 63
Figure 5.12 Local obstructions with ensemble defect 63
Figure 5.13 Motif occurrences in boxenplot 66
Figure 7.1 Empirical distribution of structure ensemble defect lower bound with minimum distance $\theta = 0$ 97
Figure 7.2 Empirical distribution of the second structure ensemble defect lower bound 99
Figure 7.3 Statistics of designable structure with minimum distance $\theta = 3$ 100
Figure 8.1 Example of dependency graph and tree decompsition 108
Figure 8.2 Example automaton for InfraRed 117
Figure 9.1 Simple example to illustrate RNAPOND problem 120
Figure 9.2 RNAPOND workflow 122
Figure 9.3 Gadget for Consistency NP-hardness proof 124
Figure 9.4 Success matrix on AntaRNA/RFAM dataset 128
Figure 9.5 Result analysis on AntaRNA/RFAM dataset 128
Figure 9.6 Success matrix on EteRNA dataset 128
Figure 9.7 Case study – EteRNA puzzle 37 129
Figure 9.8 Case study – EteRNA puzzle 58 130
Figure 9.9 Case study – EteRNA puzzle 22 130
Figure 10.1 Ensemble defect upper bound 135
Figure 10.2 Example of potential design approach 135

LIST OF TABLES

Table 2.1 Structure distances 22
Table 5.1 Local obstructions of length up to 14 for different deisgn objectives 59
Table 6.1 Designable secondary struction proportion for different design objectives 73
Table 9.1 Command line calls for the EteRNA dataset benchmark 127

ACRONYMS

BGF Bivariate Generating Function

BP Base Pair

BPs Base Pairs

CFG Context-Free Grammar

CPD Conditional Probability Distribution

CSP Constraint Satisfaction Problem

DBP Disruptive Base Pair

DBPs Disruptive Base Pairs

DFA Deterministic Finite Automaton

DP Dynamic Programming

DSV Dyck-Schützenberger-Viennot

FPT Fixed-Parameter Trackable

MEA Maximum Expected Accuracy

MFE Minimal Free Energy

MIS Maximum Independent Set

ncBPs non-canonical Base Pairs

nt nucleotide

nts nucleotides

OGF Ordinary Generating Function

PDB Protein Data Bank

RNA Ribonucleic Acid

RNAs Ribonucleic Acids

1 INTRODUCTION

Ribonucleic Acids (RNAs) are biomolecules encoding genetic information. RNA viruses, such as coronaviruses [91], store their genetic material in the form of RNAs. Messenger RNAs (mRNAs) serve as intermediate molecules to transmit genetic messages to protein synthesis. The discovery of non-coding RNAs (ncRNAs) shows that RNAs go beyond a vehicle of genetic information and participate in numerous biological processes. Transfer RNAs (tRNAs), cloverleaf-form RNAs, link amino acids and mRNAs in protein synthesis. Ribosomal RNAs (rRNAs) are large RNAs being part of ribosomes, where protein synthesis takes place. RNAs are also involved in gene regulation, such as microRNAs (miRNAs) and riboswitches. Due to the various functions of RNAs, it is believed to be the biomaterial found at the origin of life, as stated by the RNA world hypothesis and evidenced by a large body of work [40].

RNA sequence, also called primary structure, is composed of four types of *nucleotides*, Adenine (A), Cytosine (C), Guanine (G), or Uracil (U). Each nucleotide consists of a five-carbon sugar and a nucleobase attached to the first carbon. Nucleotides are connected through a (ribose-phosphate) *backbone* formed between the third 3′ carbon of one nucleotide and the fifth 5′ carbon of another one, which brings the sequence orientation from the 5′ end to the 3′ end.

An RNA sequence folds into the secondary structure, which determines the main aspects of its conformation, by forming base pairs that are mediated by hydrogen bonds. The function of an RNA depends on its structure, which is believed to be assembled hierarchically [86]. From a sequence of A, C, G, and U, nucleotides form *canonical base pairs*, including Waston-Crick base pairs A-U, C-G [88], and Wobble base pair G-U [16] (Figure 1.1a). The C-G base pair is, in general, more solid than others because of the additional hydrogen bond. Two consecutive base pairs form a *base pair stack*, which stabilizes the secondary structure (Figure 1.2). A secondary structure is said to be pseudoknotted if a base pair exists that one and only one of its nucleotide locates within the region delimited by another base pair (Figure 1.1). It is equivalent to two crossed base pairs in the linear representation (Figure 1.1c). From now on, the secondary structure is referred to *pseudoknot-free* secondary structure. Then, unpaired regions left either remain unstructured, or form *non-canonical*, relatively weaker, base pairs to stabilize and adopt complex 3D structures. Different types of non-canonical base pairs were classified in the work of Leontis and Westhof [52]. Identical small substructures, called *RNA modules*, are found in different tertiary structures [70], which either occur in a loop, such as Kink-turns, or involve two regions, like A-minor interactions.

In this thesis, we are interested in the secondary structure for the following reasons. The secondary structure is key to determine RNA function, from which the evolution-

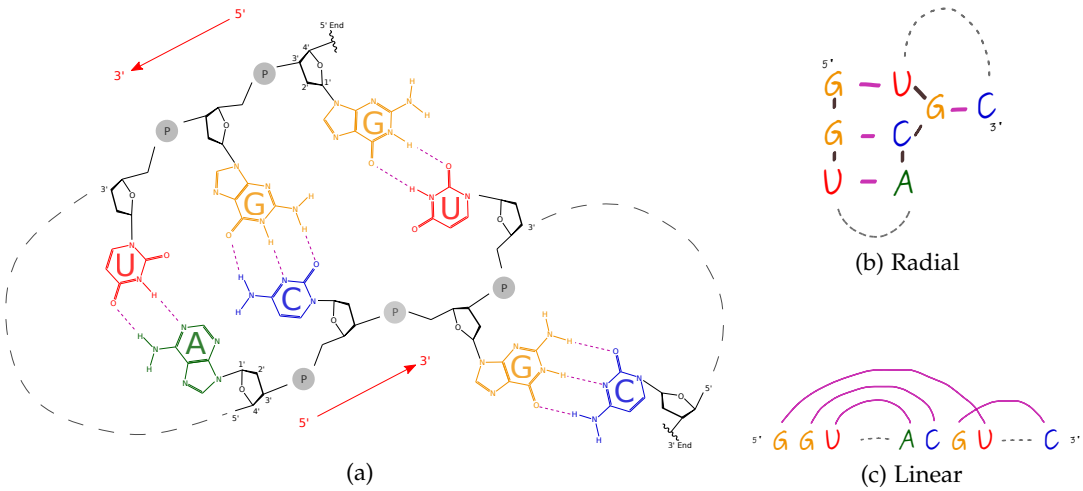


Figure 1.1: A simplified chemical form of pseudoknotted secondary structure (a) and its abstractions in a plane (b) and (c). Each nucleotide contains a nucleobase (colored) and a five-carbon sugar (black), where carbons are labeled $1'$ to $5'$. The circled P represents the (ribose-phosphate) backbone connecting $3'$ and $5'$, which defines an orientation for RNA from the $5'$ to $3'$ end. Base pairs are mediated by hydrogen bonds (purple). The radial representation (b) captures the relative positions of nucleotides, which are presented with letters. The $G-C$ base pair at the bottom right in (a) forms a pseudoknot since the nucleotide G locates in the region delimited by the base pair $G-U$, corresponding to two crossed base pairs in the linear representation (c), in which nucleotides are placed in a line.

ary pressure on RNA induced helps to identify novel RNA families [46]. It is also an essential first step towards the prediction of accurate 3D structural models [62]. In theoretical evolutionary biology, secondary structure is used to understand phenotypes and genotypes relationship [45]. Furthermore, the secondary structure can be seen as a combinatorial object [89], which facilitates computational approaches development. Having a good secondary structure prediction is then essential for designing an RNA.

RNA FOLDING PREDICTION. One of the challenges of RNA bioinformatics, resides in the prediction of RNA folding, focusing on anticipating the functional conformation adopted by an RNA. Thermodynamics is the origin of the popular nearest neighbor models [80] for RNA prediction at the secondary structure level. Each secondary structure is decomposed into smaller structures contributing additively based on individual free-energy. The energy value can be determined through experiments, Turner parameters [87], or learned from data [97]. The use of the nearest neighbor model shows good accuracy of prediction, which can be further improved by cooperating with different data sources, such as probing data [90]. Alternatively, a toy energy model, sometimes referred to as the Nussinov-Jacobson model [65], associates individual contributions to base pairs and can be useful for algorithmic design. The nature of additive energy model allows to decompose of secondary structures into several local components and solve them individually for the problem of interest.

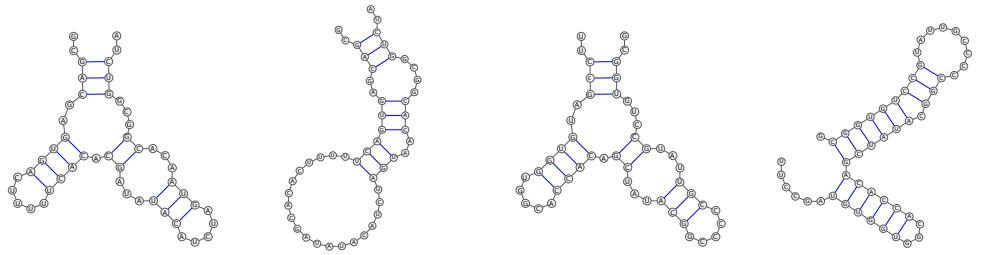
(a) Side view (b) Top view

Figure 1.2: Base pair stack in 3D view from the side (a) and the top (b). Base pairs AU and GC position in two parallel planes to reduce the exposure to the aqueous environment and nucleotides are overlapped as seen from the top view to increase the interaction between two base pairs. Both figures are drawn from RNA tridecamer [85] (PDB ID: 2R22) with PyMol [75].

In the first folding paradigm, called *MFE paradigm*, the most likely functional fold for an RNA is assumed to be its stablest one, having Minimal Free Energy (MFE) within a given energy model. The MFE structure is the structure having the lowest energy among all valid structures that given an RNA sequence. This well-defined algorithmic problem can be solved in time complexity $\mathcal{O}(n^3)$, with $\mathcal{O}(n^2)$ memory, using a dynamic programming algorithm adapted from the grammar describing secondary structures [65, 100]. Similar polynomial algorithms can also compute the energy distance between the MFE and the second most stable structure [94]. However, this energy energy difference can be very limited, or even null in the case of co-optimal structures, implying a similar probability of being observed for the MFE structure and suboptimal ones at the thermodynamic equilibrium. This drives the introduction and study of the *Boltzmann ensemble paradigm*, in which structure is associated with a Boltzmann probability related to its free-energy. Similar dynamic programming algorithms can be adapted to sample structures [22, 60] and find the most representative one(s) in the ensemble [55].

Until now, the folding prediction considers only at the thermodynamic equilibrium. However, folding is, in reality, a continuous process that passes through several structures. It can be stuck in local minima in terms of free-energy [24] or be degraded before arriving at the stationary phase [76]. This brings the studies of folding prediction in the *kinetic paradigm*. Another motivation is that some RNAs, notably, riboswitches have more than one stable conformation in the presence or absence of ligand. Thermodynamic-related paradigms and the kinetic paradigm are considered two distinct problems in RNA bioinformatics. The former ones can be solved with efficient approaches, while kinetic analyses are usually hard [56]. For this reason, especially problematic in the context of design, this thesis will focus on the secondary structure prediction at the thermodynamic equilibrium, *i.e.*, the MFE and Boltzmann ensemble paradigms.

RNA DESIGN. On the other hand, *RNA design* concerns building an RNA sequence that performs a given set of biological functions [36]. Studying the design problem



$-4.3\,\mathrm{cal}\cdot\mathrm{mol}^{-1}$ Target | $-3.8\,\mathrm{cal}\cdot\mathrm{mol}^{-1}$ Suboptimal | $-6.7\,\mathrm{cal}\cdot\mathrm{mol}^{-1}$ Target | $-16.0\,\mathrm{cal}\cdot\mathrm{mol}^{-1}$ MFE

(a) Sequence w_1 (b) Sequence w_2

Figure 1.3: Both sequences w_1 (a) and w_2 (b) fold into the target structure (left-hand side). In the context of positive design, we prefer w_2 to w_1 as the target design candidate since the free energy is lower. However, unlike the sequence w_1, the stablest (Minimal Free Energy) folding of the sequence w_2 is different than the target one. Therefore, we prefer w_1 to w_2 in the context of negative design.

are motivated by its applications in various RNA domains, such as RNA synthetic biology [81] and RNA therapeutics [93]. For example, designing artificial non-coding RNAs to control gene expression [68]. In this thesis, we are interested in RNA structural design problem, in which the secondary structure is viewed as a model of functions. More complex applications of design require the simultaneous consideration of multiple structures, such as designing artificial riboswitches as a biosensor targeting different conformations with or without ligand binding [28].

There exists two main design paradigms for design objectives (Figure 1.3). In the *positive design*, we aim to optimize the sequence affinity to the set of biological functions. One typical goal in the structural context is to design sequences to achieve minimum free-energy of the target structure. *Negative design* requires the sequences to be specific to the targets, *i.e.*, to avoid undesired functions. For instance, in structural design, unwanted structures represent an exponential number of alternative foldings different from the target ones [89].

For many types of negative objectives, this requires the well-defined adoption of a precise target structure as its predicted secondary folding. It is easy to see that a brute force method that folds all RNA sequences is unrealistic, as the expected number of secondary structures compatible with a sequence of length n can be shown to grow exponentially fast as n increases. Furthermore, the RNA design problem has been proven NP-complete in the Nussinov-Jacobson model [8]. The recent mono-structure design algorithms share similar strategy, random *seed* sequence sampling followed by an optimization step to achieve, in different folding paradigms, the MFE, high Boltzmann probability [41], or the minimum expected distance, called *ensemble defect* [96].

ENUMERATING DESIGNABLE STRUCTURES. Beyond the design of a single active RNA molecule, it is natural to ask the question, which we aim to answer in this

thesis, *How many designable secondary structures exist regarding different negative design goals?* Also, *is the structural designability a local property?* Such locality is expected because of the additivity of energy model. Indeed, with the Turner nearest-neighbor model [58], Aguirre-Hernández, Hoos, and Condon [1] discovered two undesignable motifs such that, for any sequence, there exists an alternative folding with lower free-energy. Similarly, Haleš *et al.* characterized two types of undesignable motifs for the simpler Nussinov energy model [34]. However, the impact of undesignable motifs on the structure space, and their combinatorial consequences have never been systematically studied

Structures featuring these local obstructions are undesignable as an alternative folding is always preferable locally. Enumerating secondary structures avoiding such local obstructions sets an upper bound for designable structures as structures. The infinite monkey theorem, a monkey can almost surely type any given text by randomly typing for an unlimited time, suggests that local obstruction occurs with a probability of one when the structure length is large enough. In other words, the proportion of designable structures decreases and converges to zero, but *how fast is this decay? Can it be described asymptotically?* To answer the question, we use the classic analytic combinatorics methods [30], which have been adopted to study RNA secondary structure properties [38, 53, 67, 79].

FROM POSITIVE TO NEGATIVE DESIGN. The main objective in the positive structural design is to design RNA sequence that optimizes the target structure free-energy. As implemented in INFO-RNA [12], sequence minimizing target free-energy is obtained via a dynamic programming approach. Such optimal sequence has been used as an initialization strategy for negative design. IncaRNAtion [69] is the first method to sample suboptimal sequences from a Boltzmann weighted distribution for single target design. RNAblueprint [35] uniformly samples sequences that are compatible with multiples target structures. Their successor, RNARedPrint [37], generalizes the approach for multi-target design with a structural decomposition minimizing the size of subproblem. However, alternative structures may still be adopted by the designs due to the absence of explicit negative objectives. Those structures typically differ from the target only on a local level, adopting unwanted local structures, usually base pairs, form while folding sampled sequences.

In this thesis, we investigate whether if *the negative design objective can be accomplished by preventing these disruptive base pairs during sampling.* A post-sampling sequence rejection step is inappropriate since the probability of a sequence to naturally avoid every Disruptive Base Pair (DBP) appears empirically abysmal in many cases. An alternative solution is to integrate the notion of DBPs into the framework of RNARedPrint and formalize the sampling problem as a Constraint Satisfaction Problem (CSP). It raises the following questions: *what is the proper strategy to select DBPs? Is there an RNA sequence satisfying the target structure with the given DBP set?*

PLAN OF THIS THESIS. After this brief introduction, we present, in the first part, the required notions and concepts from RNA bioinformatics to analytic combinatorics to answer the questions posed above.

- In Chapter 2, we start by formally introducing the definition of RNA secondary structure. Next, we present computational approaches to predict secondary structure at the thermodynamic equilibrium, MFE and Boltzmann ensemble paradigms. We illustrate the dynamic programming algorithms employed in each method with a simple base pair energy model.
- We define in Chapter 3 the objectives of positive and negative structural design, followed by stating the negative RNA design problem that we are focusing on in this thesis. We also present some state-of-the-art methods for both positive and negative design.
- Chapter 4 introduces the basic notions in language theory and analytic combinatorics. We present generating function as a means to describe combinatorial class properties and then as a function for further analyses. We use RNA secondary structure as an example to illustrate the concept in this chapter.

In the second part of this thesis, we study the RNA design problem from the perspective of combinatorics. We present two approaches to estimate upper bounds for designable secondary structures.

- In Chapter 5, we first introduce the notions related to the local motifs with an extension of structural concepts. Then, we identify undesignable motifs for different negative design goals. We also investigate identified local obstructions in experimentally determined structures.
- Using the fact that a structure is not designable if one of its local decompositions is not, we propose a grammar to enumerate secondary structures avoiding undesignable motifs in Chapter 6. We then show that the proportion of designable secondary structures decreases exponentially with the number of nucleotides
- Chapter 7 introduces another approach to compute the upper bound for designable structures. We introduce a lower bound on the ensemble defect which is shown to follow a Normal limiting distribution with mean and variance linear to the number of nucleotides. The upper bound is then the cumulative distribution function evaluated at the value determined by the design objective.

Last but not least, in the third part of the thesis, we show the possibility of extending the state-of-art positive design algorithm into a promising solution to the negative design.

- We present InfraRed, a generalization of the RNARedPrint framework in Chapter 8. We describe the problem considered and present the approaches used in this novel sampling framework. As a usage application, we show a reimplementation of the IncaRNAtion algorithm with our framework and the integration of automata.

- In Chapter 9, we introduce the notion of Disruptive Base Pairs (DBPs), and their inclusion within the InfraRed framework. We show that determining if a compatible sequence exists is an NP-hard problem, which can nevertheless be solved in polynomial time on instances of bounded treewidth. Then, we present RNAPOND, a negative design tool with an iterative sampling strategy using InfraRed, which shows comparable performance with state-of-the-art methods.

Chapter 10 summarizes the works presented in this thesis. We also discuss possible further research directions.

Part I

BACKGROUND KNOWLEDGE

2

RNA 2D STRUCTURE PREDICTION

2.1 NOTION RELATED TO RNA BIOINFORMATICS

This section introduces the notions concerning RNA secondary structure that we will use in this work.

An RNA a sequence of n nucleotides, Adenine (A), Cytosine (C), Guanine (G), or Uracil (U).

Definition 2.1 (RNA sequence): An RNA sequence w of length n is a string $w_1 \cdots w_n$, each w_i taking value from $\Sigma = \{A, C, G, U\}$ is the nucleotide, also called base, at position i, *i.e.* $w \in \Sigma^n$. Its length is denoted by $|w|$.

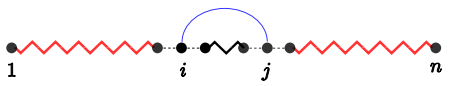
Definition 2.2 (Base Pair): A base pair (i, j) with $i \neq j$ is a pair of nucleotides at positions i and j. The position j is said to be the partner of i.

In this work, we consider a base pair as a collection of nucleotides, *i.e.*, $(i, j) = (j, i)$.

Definition 2.3 (Secondary structure): A *secondary structure* of length n is a set S of base pairs (i, j), $1 \leqslant i < j \leqslant n$ such that,

1. Each position is involved in at most one base pair;
2. Base pairs are pairwise non-crossing, $\nexists (i, j), (k, l) \in S$, $i < k < j < l$;
3. *Minimal distance* of θ is between paired positions, $\forall (i, j) \in S$, $j - i > \theta$.

Figure 2.1 shows examples where three conditions are dissatisfied. Structures having crossed base pairs, *i.e.*, dissatisfying the second condition, are called *pseudo-knotted*, which are not considered in this thesis. Furthermore, a region $[i, j]$ delimited by the base pair (i, j) is independent to the region outside $[1, i-1] \cup [j+1, n]$ of (i, j) as no base pair are allowed to be formed across two regions. This property turns to be a key component while adapting Dynamic Programming (DP) algorithm on secondary structure, where the delimited region $[i, j]$ can be treated as a subproblem.



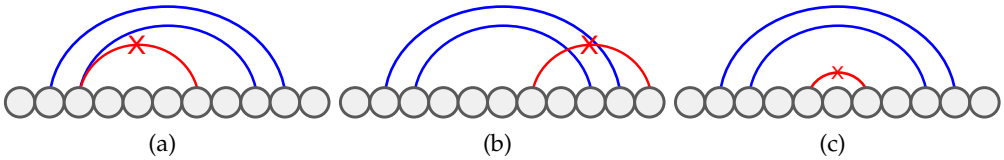


Figure 2.1: Invalid secondary structures with minimum distance $\theta = 3$ due to the presence of base pair in red corresponding, respectively, to three conditions in Definition 2.3. (a) Base at position 3 has two partners; (b) Base pairs in red and blue are crossing, forming a pseudoknot; (c) For the classic $\theta = 3$ parametrization, the red base pair involves positions at insufficient base.

Definition 2.4 (Unpaired nucleotide): Unpaired nucleotides of a secondary structure S of length n is the set of positions $\{i \in [1, n]; \forall j \neq i, (i, j) \notin S\}$ that are not involved in base pairs.

At the secondary level, the nucleotide contents of a valid base pair, called a *canonical base pair*, is either in $\{(A, U), (C, G), (G, C), (U, A)\}$ (a Waston-Crick base pair) or in $\{(G, U), (U, G)\}$ (a Wobble base pair).

Definition 2.5 (Compatible sequence and secondary structure): Let w be a sequence of length n and S be a secondary structure of length n. We say w and S are compatible if for all base pairs $(i, j) \in S$, nucleotides w_i and w_j form a canonical base pair.

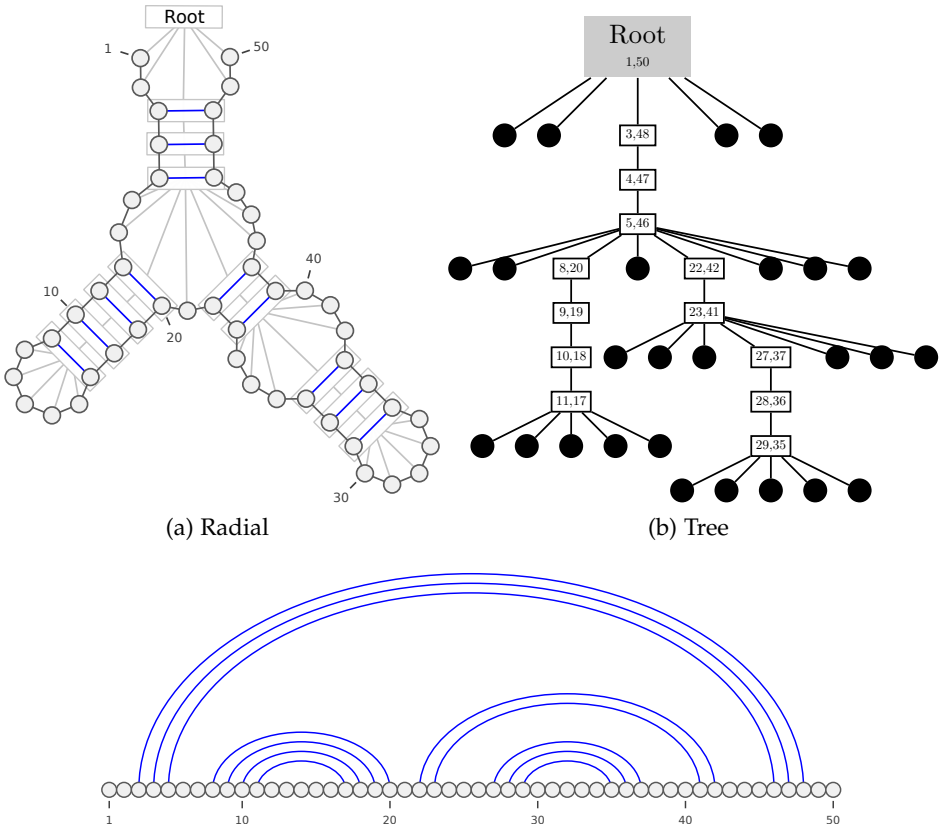
From now on, we use the following notations to denote different sets of secondary structures

- $\mathcal{S}^\theta$ for the set of secondary structures with minimum distance θ;
- $\mathcal{S}^\theta_n$ the restriction on length n, $\mathcal{S}^\theta_n := \{S \in \mathcal{S}^\theta; |S| = n\} \subset \mathcal{S}^\theta$;
- $\mathcal{S}^\theta_w \subseteq \mathcal{S}^\theta_{|w|}$ for the set of secondary structures compatible with sequence w.

In the absence of ambiguity, θ is omitted from the notation.

A secondary structure can be presented in different ways, as shown in Figure 2.2.

- **Radial representation.** Structure is presented as a graph with vertices for nucleotides and edges for backbones (black) or base pairs (blue). Consecutive base pairs are drawn in a ladder-like shape while unpaired nucleotides are positioned as a circle between paired regions.
- **Tree representation.** A secondary structure S can be unambiguously represented as a rooted ordered tree $T = (V := V_i \cup V_l, E)$, whose internal vertices are intervals $[i, j] \in V_i$, $i < j$, representing base-paired positions (i, j) in S, and leaves are singletons $\{i\} \in V_l$, representing an unpaired position i in S. Any edge $(u \to v) \in E$, *i.e.*, v is the parent of u, connects intervals such that $u \subset v$ and $\nexists v' \in V_i$ such that $u \subset v' \subset v$.



(a) Radial (b) Tree

(c) Linear

••(((••((((•••••))))•((•••(((•••••)))•••))•••)))••

(d) Dot-Bracket notation

Figure 2.2: Representations of secondary structure, radial (a), tree (b), linear (c), and dot-bracket notation (d). In radial representation, virtual lines in light gray are added in the graph to show the parent-children relationship for tree representation.

- **Linear representation.** Nucleotides are positioned in a line, and each base pair i, j is presented by an arc connected positions i and j in the upper half-plane.
- **Dot-Bracket notation.** Structures can be represented as a well-parenthesized expression, *i.e.*, a word of the language formed from three letters `(`, `)` (paired bases), and • (unpaired base), such that the number of `(` and `)` are equal, and all prefixes have more `(` than `)`.

Radial and linear representation can be drawn using VARNA [18], an RNA visualization tool, with secondary structure in dot-bracket notation as input.

2.2 SECONDARY STRUCTURE PREDICTION WITH ENERGY MINIMIZATION

Thermodynamics is, at the origin, a popular model for RNA prediction at the secondary structure level. At the thermodynamics equilibrium, the MFE structure is the

most likely to be adopted by an RNA, and it is therefore a natural candidate for its functional fold.

Definition 2.6 (Minimal Free Energy Structure): Given an energy model $\mathcal{E}$ and a sequence w, the Minimal Free Energy (MFE) structure(s) $\mathsf{MFE}(w)$ of w is a set of secondary structures such that

$$\mathsf{MFE}(w) = \{S \in \mathcal{S}_w;\ \mathcal{E}(w, S) = \min_{S' \in \mathcal{S}_w} \mathcal{E}(w, S')\}.$$

2.2.1 *Base Pair Energy Model*

The first non-exponential time MFE folding algorithm is proposed by Nussinov and Jacobson [65] while considering a base pair energy model, or sometimes referred to as the Nussinov-Jacobson model. The model assumes that the most stable structure is the one having the most base pairs. This assumption is equivalent to considering that structure energy is uniquely and equally contributed from base pairs, for example, -1 kcal.mol^{-1} per valid base pair regardless of its nucleotides content. Given a sequence w of length n, finding the with secondary structure while maximizing base pairing is then identical to finding the MFE structure compatible with w.

The approach adopts a Dynamic Programming (DP) scheme based on the structure decomposition [89], as shown in Figure 2.3. In the region between positions i and j, the i-th nucleotide is either unpaired or paired with another nucleotide at position k. For the former case, we define $M_{i,j}$ to represent the minimum energy within the region $[i+1, j]$. As for the later one, base pair (i, k) splits $[i, j]$ into two independent regions $[i+1, k-1]$ and $[k+1, j]$ with minimum energy $M_{i+1,k-1}$ and $M_{k+1,j}$. Thus, the minimum energy $M_{i,j}$ between positions i and j is

$$M_{i,j} = \min \begin{cases} M_{i+1,j} \\ \min\limits_{i<k\leqslant j} \Delta G(i,k) + M_{i+1,k-1} + M_{k+1,j} \end{cases} \tag{2.1}$$

where$\Delta G(i, k)$ is the base pair free energy of (i, k), indicating whether nucleotides w_i and w_k can form a base pair,

$$\Delta G(i,k) = \begin{cases} -1 & \text{if } (w_i, w_k) \in \{(A,U), (C,G), (G,C), (G,U), (U,A), (U,G)\} \\ +\infty & \text{otherwise.} \end{cases}$$

A DP algorithm is then used to compute $M_{1,n}$, the minimum energy achieved given sequence w, followed by backtracking to obtain the secondary structure whose energy is $M_{1,n}$. At each region $[i, j]$, the backtracking Backtrack decides whether i-th

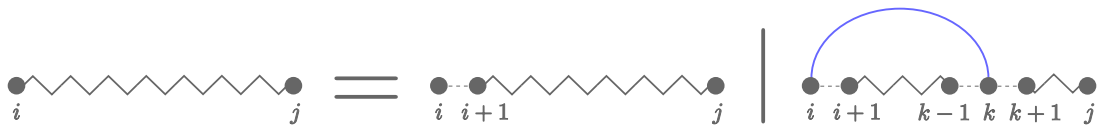


Figure 2.3: Classic secondary structure decomposition. The base i is either unpaired or paired with a base $k \in [i+1, j]$. Notice that, in the original version used by Stein and Waterman [77], the decomposition was started by the base j. These two decompositions are equivalent. Here, we start by the base i to be consistent with other decompositions used. In addition, the decomposition is an unambiguous version of the Nussinov decomposition [65].

nucleotide is unpaired or paired with a nucleotide at position k, which can be expressed recursively as

$$\mathsf{Backtrack}(i,j) = \begin{cases} \mathsf{Backtrack}(i+1,j) & \text{if } M_{i,j} = M_{i+1,j} \\ \{(i,k)\} \cup \mathsf{Backtrack}(i+1,k-1) \cup \mathsf{Backtrack}(k+1,j) \\ \qquad \text{if } M_{i,j} = \Delta G(i,k) + M_{i+1,k-1} + M_{k+1,j}, i < k \leqslant j \end{cases}$$

with the base case $\mathsf{backtrack}(i,j) = \varnothing$ when $i > j$. The total algorithm complexity is $\mathcal{O}(n^3)$ in time and $\mathcal{O}(n^2)$ in space.

The same algorithm can be used with a more general base pair energy model, where different energy values are assigned to base pairs depending on nucleotides. In fact, (C, G) and (G, C) base pairs are generally more stable than others.

2.2.2 *Energy Model with Loop Decomposition*

In a more realistic energy model, the structure energy is assumed to be the additive energy contributions from basic units, called *loops* or *shallow subtrees* in tree representation, and their associated nucleotides.

A shallow subtree is a subtree of depth 1, *i.e.*, an internal vertex with its first-generation descendants. Note that the definition of a subtree in this section differs from the usual one in graph theory, where a subtree includes all node descendants. The base pair at the root of a subtree is called the *closing base pair* since it encloses the loop, in which the enclosed base pairs are called *open base pairs*. Figure 2.4 shows a decomposition of secondary structure into different types of loop determined by the number of open base pairs k.

- **Hairpin.** An unpaired region delimited by a base pair, *i.e.*, $k = 0$.
- **Base pair stack.** A loop consists of two consecutive base pairs (i,j) and $(i+1, j-1)$. Several consecutive stacks form a *helix*. Starting from base pair (i,j), a helix of length l is composed of base pairs $(i,j), \ldots, (i+l-1, j-l+1)$.
- **Internal loop.** The closing base pair encloses two strands of unpaired nucleotides separated by an open base pair $(k = 1)$. If the length of one strand is null, the internal loop is also called a *bulge*.

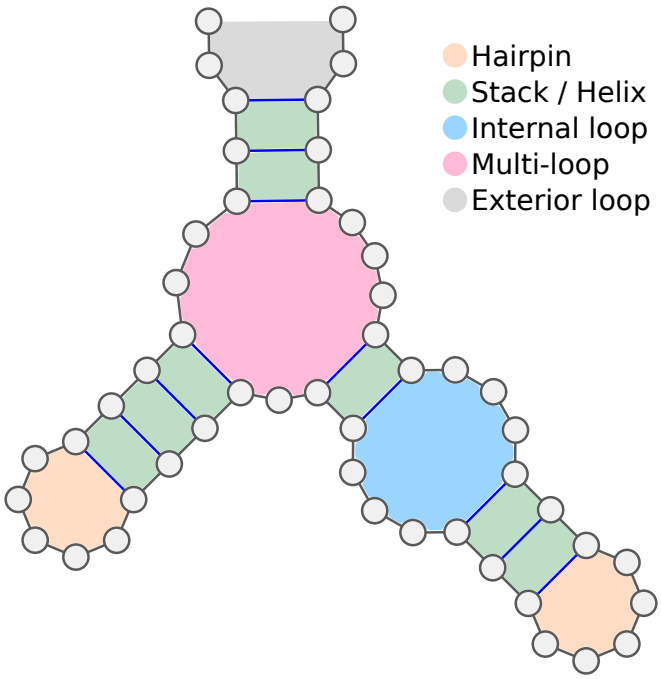


Figure 2.4: Loop decomposition of a secondary structure, including hairpins (orange), stacks (green), an internal loop (blue), a multi-loop (red), and an exterior loop (gray).

- **Multi-loop, or multibranch loop.** A multi-loop contains more than one open base pair ($k \geqslant 2$), or called branch in multi-loop. The open base pair amount is also called the in multi-loop.

In addition, the region delimited by the virtual root is sometimes called the *exterior loop*.

Now, we can formulate a formal definition for the energy model based on loop decomposition.

Definition 2.7 (Energy Model): An *energy model* $\mathcal{E}$ taking a sequence w and a secondary structure S of same length is a function $\mathcal{E} : \Sigma^* \times \mathcal{S} \to \mathbb{R} \cup \{+\infty\}$ such that

$$\mathcal{E}(w, S) = \sum_{T = \overset{p}{\underset{a\ b\ c\ \cdots}{\wedge}} \in S} \Delta G\left(\{p \to w_p, a \to w_a, b \to w_b \ldots\}, T\right)$$

where $\Delta G\,(m, T)$ is the free-energy, expressed in kcal.mol^{-1} associated with the assignment m of concrete nucleotides from w to the (pairs of) positions in the subtree T.

One of the most used energy models is the Turner nearest-neighbor model or Turner energy model [87]. The energy model offers a database of energy parameters experimentally determined for small loops. For large loops, energy is extrapolated using a closed-form expression relying on the type, the length of the loop, and the

nucleotides nearby. Secondary structure energy is then the sum of loop energies using provided parameters.

ZUKER ALGORITHM Zuker and Stiegler [99] proposed a DP algorithm to compute the MFE structure using loop decomposition based energy model. The algorithm uses a DP scheme derived from an extension of the classic structure decomposition. In the extended version, the existence of closing base pair is taken into account to determine the loop type for further decomposition. More precisely, the loop delimited by base pair (i, k) in the classic decomposition (Figure 2.4) has three possible loop types depending on the decomposition on the region $[i+1, k-1]$.

1. A hairpin if $[i+1, k-1]$ is an unpaired region;
2. An internal loop if a base pair (i', j') forms in $[i+1, k-1]$. Because of two novel variables, the complexity of energy minimization is $\mathcal{O}(n^4)$ in time. In practice, it is reduced to $\mathcal{O}(n^3)$ by restricting strand length;
3. A multi-loop if at least two base pairs form in $[i+1, k-1]$. In principle, no efficient decomposition exists for multi-loop due to the uncertain branch number. To workaround, Turner model assumes the energy of a multi-loop grows linearly with branch number and average asymmetry, which allows decomposing multi-loop by adding at most one open base pair each time.

Despite the exponential number of secondary structures, the energy minimization is achieved in $\mathcal{O}(n^3)$ with DP algorithm proposed by Zuker and Stiegler [99]. The approach is further extended to compute all secondary structures with free-energy within a range from MFE [94] and implemented in the library ViennaRNA.

Due to the complexity of Zuker structure decomposition, we will use the base pair energy model with a minimum distance $\theta = 0$ to illustrate other RNA-related algorithms for the rest of this chapter. Similar algorithms can be extended for Turner energy model by changing the structure decomposition.

2.3 BOLTZMANN DISTRIBUTION PARADIGM

At the thermodynamic equilibrium, secondary structure can be observed with the probability related to its free-energy. The MFE structure has the highest probability of being observed, while unstable structures are expected to have minimal probabilities. In this paradigm, structures with free-energy closed to the MFE have a similar probability to the MFE structure. It is then also interesting to obtain these suboptimal structures.

Under the hypothesis of a Boltzmann equilibrium, the structure in the ensemble follows the Boltzmann distribution.

Definition 2.8 (Boltzmann Probability): For a given sequence w, the putative secondary structure S follows a Boltzmann distribution

$$\mathbb{P}(S \mid w) = \frac{\mathcal{B}(w, S)}{\mathcal{Z}_w}$$

with

$$\mathcal{B}(w, S) := e^{-\frac{\mathcal{E}(w,S)}{RT}} \qquad \text{is the } \textit{Boltzmann factor} \text{ of } w \text{ and } S$$

$$\mathcal{Z}_w := \sum_{S' \in \mathcal{S}_{|w|}} \mathcal{B}(w, S') \qquad \text{is the } \textit{partition function} \text{ of } w$$

where R is the Boltzmann constant and T is the temperature.

2.3.1 *Partition Function Computation.*

Computing partition function is the key to access these probabilities. The number of secondary structures grows exponentially, so an explicit sum would be unfeasible beyond several dozen nucleotides. Fortunately, the partition function can also be calculated in polynomial time on the length, using a DP algorithm [60]. Analog with Equation 2.1, McCaskill algorithm uses the same DP scheme for energy minimization with a change of algebra, operations $(+, \times)$ substitutes $(\min, +)$. The additivity of free-energy ensures that multiplying the contributions from smaller regions gives the partition function.

More precisely, given a pair of positions (i, j), the goal is to compute the partition function defined over the region $[i, j]$,

$$\mathcal{Z}_{i,j} = \sum_{S' \in \mathcal{S}_{j-i+1}} \mathcal{B}(w_i \cdots w_j, S').$$

Replacing the energy by the Boltzmann factor, Equation 2.1 becomes,

$$\mathcal{Z}_{i,j} = \mathcal{Z}_{i,j-1} + \sum_{i<k\leqslant j} e^{-\frac{\Delta G(k,j)}{RT}} \mathcal{Z}_{i,k-1} \mathcal{Z}_{k+1,j-1}$$

with $\mathcal{Z}_{i,j} = 1$ if $i > j$ for the base case. The total partition function $\mathcal{Z}_w = \mathcal{Z}_{1,n}$ is computed starting with the entire sequence with the complexity $\mathcal{O}(n^3)$ in time and $\mathcal{O}(n^2)$ in space.

2.3.2 *Structure Sampling*

Ding and Lawrence [22] proposed an algorithm generating a valid structure from the ensemble with respect to its Boltzmann probability. It performs stochastic backtracking after computing the partition function. Unlike the backtracking used for energy

minimization, for the region $[i, j]$, stochastic backtracking selects a partner k for nucleotide i according to the probability of having base pair (i, k) within the region $[i, j]$.

More precisely, the probability of having base pair (i, k) in a random structure is determined by the region inside $[i+1, k-1]$ and outside $[k+1, j]$ of the base pair (i, k),

$$\mathbb{P}^{[i,j]}(i,k) = \frac{e^{-\frac{\Delta G(i,k)}{RT}} \mathcal{Z}_{i+1,k-1} \mathcal{Z}_{k+1,j}}{\mathcal{Z}_{i,j}}$$

where partial partitions $\mathcal{Z}_{i,j}$, $\mathcal{Z}_{i+1,k-1}$, and $\mathcal{Z}_{k+1,j}$ are precomputed while computing the total partition function $\mathcal{Z}_{1,n}$. On the other hand, the probability of position i being unpaired, or pairing to itself, is then

$$\mathbb{P}^{[i,j]}(i,i) = 1 - \sum_{i<k\leqslant j} \mathbb{P}^{[i,j]}(i,k).$$

Therefore, the partner $k \in [i, j]$ for nucleotides i is selected according to the conditional probability $\mathbb{P}^{[i,j]}(i, k)$.

In practice, the division by the total partition function is skipped since the denominator is the same for all probabilities. Let $N_i, \ldots, N_j$ be the nominators of probabilities $\mathbb{P}^{[i,j]}(i,i), \ldots, \mathbb{P}^{[i,j]}(i,j)$ with $\mathcal{Z}_{i,j} = N_i + \cdots + N_j$. The partner k is then selected such that

$$\sum_{l=i}^{k} N_l \leqslant x < \sum_{l=i}^{k+1} N_l$$

with x is a random value uniformly chosen from $[0, \mathcal{Z}_{i,j}[$. It is achieved by subtracting N_l from x for l from i to j until x becomes negative.

Sampling k sequences of length n needs $\mathcal{O}(kn^2)$ time complexity for the worst case and $\mathcal{O}(kn\sqrt{n})$ for the average case, which can be improved to $\mathcal{O}(kn\sqrt{n})$ for both with Boustrophedon strategy [66]. For some applications, it requires preventing the sampling of structures that have been seen. On top of stochastic backtracking, one can achieve non-redundant sampling while keeping the same distribution by subtracting from partial partition functions, the contribution of each sampled structure [63].

2.3.3 *Base Pair Probability*

McCaskill [60] also provided a method, called the inside-outside algorithm, to compute the probability of observing a base pair in the ensemble with the same principle of partition function computation.

Definition 2.9 (Base Pair Probability): For a given sequence w, the probability of a base pair (i, j) is defined as

$$p_w(i,j) = \sum_{\substack{S \in \mathcal{S}_n \\ (i,j) \in S}} \mathbb{P}(S \mid w).$$

In the case of $i = j$, $p_w(i,i)$ represents the probability of i being left unpaired, which is usually denoted by q_i,

$$q_i = p_w(i,i) = 1 - \sum_{j \neq i} p_w(i,j).$$

Let w be a sequence and (i, j) be a base pair. As seen on the left-hand side of Figure 2.5, sequence is divided into two regions, the inside one $[i+1, j-1]$ and the outside one $[1, i-1] \cup [j+1, n]$. The partition function with the base pair (i, j) is then the product of contributions from outside and inside regions. The contribution from the inside region to the partition function is exactly the partition function $\mathcal{Z}_{i+1,j-1}$ since the base pair delimit the region. On the other hand, the outside contribution is not simply $\mathcal{Z}_{1,i-1} \times \mathcal{Z}_{j+1,n}$ since a base in $[1, i-1]$ can form a base pair with a base in $[j+1, n]$. Let $\mathcal{Y}_{i,j}$ be the contribution from the region outside the base pair (i, j). The base pair probability is

$$p_w(i,j) = \frac{e^{-\frac{\Delta G(i,k)}{RT}} \mathcal{Y}_{i,j} \mathcal{Z}_{i+1,j-1}}{\mathcal{Z}_w}.$$

To compute $\mathcal{Y}_{i,j}$, the decomposition starts from the inside of sequence to the outside. As presented in the right-hand side of Figure 2.5, there are three situations on base $i-1$.

- Base $i-1$ is unpaired. The outside region is reduced to $[1, i-2] \cup [j+1, n]$;
- Base $i-1$ paired to a base i' in $[1, i-2]$. The outside region is reduced to $[1, i'-1] \cup [j+1, n]$ while an independent region $[i'+1, i-2]$ is introduced;
- Base $i-1$ paired to a base j' in $[j+1, n]$. The outside region is reduced to $[1, i-1] \cup [j'+1, n]$ while an independent region $[j+1, j'-1]$ is introduced.

Thus,

$$\mathcal{Y}_{i,j} = \mathcal{Y}_{i-1,j} + \sum_{1 \leqslant i' < i-1} e^{-\frac{\Delta G(i',i-1)}{RT}} \mathcal{Y}_{i',j} \mathcal{Z}_{i'+1,i-1} + \sum_{j \leqslant j' < n} e^{-\frac{\Delta G(i-1,j')}{RT}} \mathcal{Y}_{i,j'} \mathcal{Z}_{j+1,j'-1}$$

with $\mathcal{Y}_{i,j} = 1$ if $i > j$ for the base case.

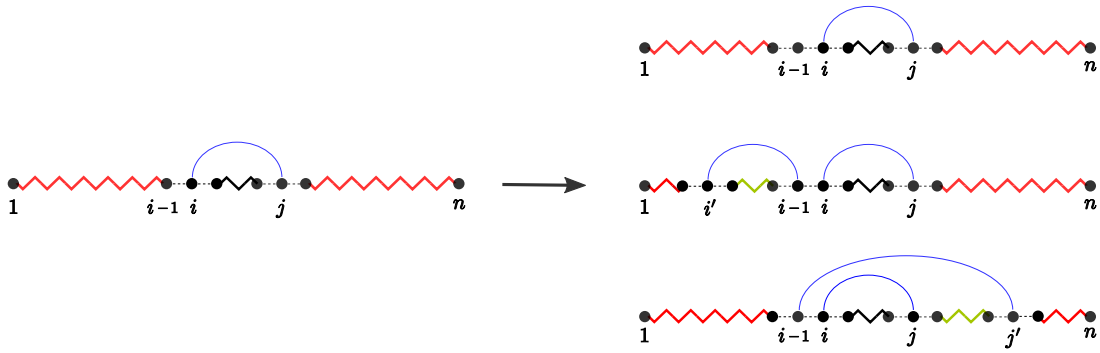


Figure 2.5: Structure decomposition outside of base pair (i, j) from base $i-1$. Outside region is marked by red and the newly introduced inside region after the decomposition is marked by green.

2.4 ENSEMBLE REPRESENTATIVES AND EXPECTED DISTANCE

Under the thermodynamic hypothesis, the MFE secondary structure is the most stable and achieves the highest Boltzmann probability. While considering the entire structure ensemble, competitive structures with probability close to the MFE can be far in the ensemble while surrounding structures have a poor probability. In such a case, it is preferable to look for the representative structure(s) within the ensemble, rather than the MFE.

2.4.1 *Structure Distance*

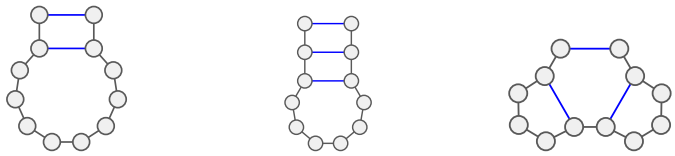
To define the concept of representative structure, we need to introduce a notion of distance between two structures, $\text{dist} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_+$.

An intuitive solution is to consider the Hamming distance, *i.e.*, the number of differing positions, between the two representations of structures as well-parenthesized strings. However, the Hamming distance does not take the paired partner into account, as shown in Table 2.1. Two alternative distances have been used. Let S_1, S_2 be two secondary structures of equal length.

- **Base Pair Distance (**$\mathtt{BPdist}$**)** is the number of base pairs in one structure but not in the other,

  $$\mathtt{BPdist}(S_1, S_2) = |S_1 \Delta S_2|$$

  with Δ denotes the symmetric difference between two sets of base pairs, S_1, S_2. Base pair distance is also the minimum number of base pairs needed to add/delete from S_1 to S_2.



S_1 : ((••••••••)) S_2 : (((••••••))) S_3 : ((•••)(•••))

	Hamming Dist.	Base Pair Dist.	Diff. Paired Dist.
S_1	((••••••••))	((••••••••))	((••••••••))
S_2	(((••••••)))	(((••••••)))	(((••••••)))
Distance	2	$0+1=1$	2
S_1	((••••••••))	((••••••••))	((••••••••))
S_3	((•••)(•••))	((•••)(•••))	((•••)(•••))
Distance	2	$1+2=3$	4

Table 2.1: Different structure distances for S_1, S_2 and S_1, S_3 with nucleotides counted in the distance are colored. Two colors are used for base pair distance since base pairs of both structures contribute to the distance. In both case, the Hamming distance between two structures is 2. However, it is clear that S_1 is closer to S_2 than to S_3 from structural aspect. The difference is both captured using base pair distance and difficulty paired distance.

- **Differently Paired Distance (**DPdist**)** is the number of nucleotides paired differently in both structures. Let T_k be the partner sequence of $S_k, k \in \{1, 2\}$,

$$T_k(i) = \begin{cases} j & \text{if base } i \text{ paired with } j \text{ in } S_k \\ i & \text{if base } i \text{ unpaired in } S_k. \end{cases}$$

  Then, the differently paired distance of S_1 and S_2 is the hamming distance of T_1 and T_2.

The most representative structure, also called centroid solution, is the secondary structure having the minimum expected distance to a random structure in Boltzmann distributed ensemble, *i.e.*, structure at the center of the ensemble.

Definition 2.10 (Centroid Solution): Let w be a sequence of length n and

$$S = \operatorname*{argmin}_{S \in \mathcal{S}_n} \sum_{S' \in \mathcal{S}_n} \mathbb{P}(S' \mid w) \cdot \mathsf{dist}(S', S).$$

2.4.2 *Maximum Expected Accuracy*

Considering the differently paired distance, one has

$$\sum_{S' \in \mathcal{S}_n} \mathbb{P}(S' \mid w) \cdot \mathrm{DPdist}(S', S) = n - \sum_{(i,j) \in S} 2p_w(i,j) - \sum_{i \text{ unpaired in } S} q_i.$$

The structure minimizing the expected distance is then the structure maximizing the expected accuracy,

$$\sum_{(i,j) \in S} 2p_w(i,j) + \sum_{i \text{ unpaired in } S} q_i.$$

Lu, Gloor, and Mathews [55] proposed a DP algorithm to compute the MEA structure with an additional parameter γ for base pair probability.

$$\mathsf{EA}(S) = \sum_{(i,j) \in S} \gamma \cdot 2p_w(i,j) + \sum_{i \text{ unpaired in } S} q_i.$$

Given a sequence w of length n, the MEA can be calculated using the same structure decomposition. Let $\mathsf{MEA}_{i,j}$ be the MEA between positions i and j. One has, similar to Equation 2.1,

$$\mathsf{MEA}_{i,j} = \max \begin{cases} q_i + \mathsf{MEA}_{i+1,j} \\ \max\limits_{\substack{i<k\leqslant j \\ (w_i, w_k) \in \mathcal{B}}} \gamma \cdot 2p_w(i,k) + \mathsf{MEA}_{i+1,k-1} + \mathsf{MEA}_{k+1,j} \end{cases}$$

with $\mathsf{MEA}_{i,j} = 0$ if $i > j$. The MFE given a sequence is computed with region $[1, n]$ and the MFE structure is obtained via backtracking.

3 RNA STRUCTURAL DESIGN

3.1 DESIGN OBJECTIVES

There are two main design paradigms for RNA structural design based on the design objectives. In the positive design, we aim to optimize the free-energy, used as a proxy for the affinity, towards a limited number of structures. In the context of negative design, it requires the sequence to be specific to the target(s), *i.e.*, to avoid folding into an exponential number of undesired structures. When more than one target structure is given, the design problem is called multi-target design.

3.1.1 *Positive Design*

Let S^* be a target secondary structure S^* of length n. Under thermodynamic hypothesis, the goal of positive design is to find sequence w^* that minimizes the target free-energy,

$$w^* := \underset{w \in \{A,C,G,U\}^n}{\operatorname{argmin}} \quad \mathcal{E}(w, S^*)$$

where $\mathcal{E}$ is an energy model, such as the Turner energy model. A particular energy model is to assign -1 to sequence compatible with the target structure and $+\infty$ to incompatible one. In this case, the goal becomes finding compatible design sequences.

Finding suboptimal sequences is often needed when several design sequences are demanded. Similar to suboptimal structure sampling mentioned in Section 2.3, design sequence w is obtained with probability

$$\mathbb{P}(w \mid S^*) \propto e^{\beta \mathcal{E}(w, S^*)}$$

where $\mathcal{E}$ is an energy model and β is an arbitrary constant. If $\beta = -1/RT$, then the sequence probability is proportional to the usual Boltzmann factor. In addition, a compatible sequence is uniformly sampled when $\beta = 0$ regardless of the energy model. It requires precomputing the *dual partition function* for sequence sampling.

Definition 3.1 (Dual Partition Function): Given a target structure S^* of length n and an energy model $\mathcal{E}$, the dual partition function $\mathcal{Z}_{S^*}$ is defined as

$$\mathcal{Z}_{S^*} = \sum_{w \in \Sigma^n} e^{\beta \mathcal{E}(w, S^*)}$$

where β is an arbitrary constant.

The dual partition function sums the Boltzmann factor over the sequence space while the usual partition function is defined on structures.

In multi-targets design, the design sequence needs to be compatible with all targets and minimize some combinations, such as linear function, of the target free-energies. For suboptimal sequence sampling, sequence w is sampled from the Boltzmann-weighted distribution such that

$$\mathbb{P}(w \mid S_1^*, \ldots, S_k^*) \propto e^{\beta \mathcal{E}(w, S_1^*)} \ldots e^{\beta \mathcal{E}(w, S_k^*)}$$

where $S_1^*, \ldots, S_k^*$ are target structures.

3.1.2 *Negative Design*

Given a target secondary structure S^*, the classic negative RNA design problem or RNA inverse folding problem, consists in producing a sequences w that adopts S^* as its unique MFE structure.

Problem 1 (RNA Inverse Folding):
Input: Target structure S^* of length n
Output: Sequence $w \in \Sigma^n$, such that

$$\mathsf{MFE}(w) = \{S^*\}.$$

Despite of being the MFE structure, alternative structures can be competitive in the ensemble, for example, structure with close Boltzmann probability. A notion of *defect* captures the avoidance of alternative structures. The smaller the defect value is, the harder it will be for the design sequence to adopt a (significantly) alternative structure.

Definition 3.2 (Defect): Given an RNA sequence $w \in \Sigma^*$ and a target structure $S^* \in \mathcal{S}$, a defect is a function

$$\begin{array}{rccccc} D: & \Sigma^* & \times & \mathcal{S} & \to & \mathbb{R} \\ & w & \times & S^* & \mapsto & D(w, S^*) \end{array}$$

computing the hardness of folding w into S^*.

The meta-objective of negative RNA design can then be summarized as:

Problem 2 (Negative RNA Design):
Input: Real-valued threshold ε, defect D, target structure S^*
Output: Sequence $w \in \Sigma^{|S^*|}$, called a *(negative)* $(D \leqslant \varepsilon)$*-design* for S^*, such that

$$\mathsf{MFE}(w) = \{S^*\} \quad \text{and} \quad D(w, S^*) \leqslant \varepsilon. \tag{3.1}$$

The first objective requires the design to adapt the target structure as its stablest conformation. The second one enforces the avoidance of competing structures in the structure ensemble induced by the design. We call $(D \leqslant \varepsilon)$-designable a secondary structure that does admit at least a valid design, and denote by $\mathcal{D}^{D \leqslant \varepsilon}$ the set of $(D \leqslant \varepsilon)$-designable secondary structures.

RNA design methods usually consider one of the three following defects, suboptimal defect, probability defect, and ensemble defect.

Definition 3.3 (Suboptimal Defect): The *Suboptimal Defect* D^S of a sequence w is defined as the energy difference to the first suboptimal, such that

$$D^S(w, S^*) := \min_{\substack{S \in \mathcal{S}_{|w|} \\ S \neq S^*}} \mathcal{E}(w, S^*) - \mathcal{E}(w, S) \in \mathbb{R}$$

where $\mathcal{E}$ is an energy model.

In practice, the value of suboptimal defect D^S is negative or null because of the first design objective, which demands the target to be the MFE structure. Thus, the design problem with objective $D^S \leqslant 0$ is equivalent to the classic inverse folding.

Definition 3.4 (Probability Defect): The *Probability Defect* D^P represents the probability of folding into any other structure than S^*:

$$D^P(w, S^*) := \sum_{\substack{S \in \mathcal{S}_{|w|} \\ S \neq S^*}} \mathbb{P}(S \mid w) = 1 - \mathbb{P}(S^* \mid w) \in [0, 1].$$

With the probability defect, the target structure is designed by optimizing the Boltzmann probability.

Definition 3.5 (Ensemble Defect): The *Ensemble Defect* D^E is the expected amount of bases differently paired between S^* and a random structure, generated with respect to the Boltzmann probability distribution:

$$D^E(w, S^*) := \sum_{S \in \mathcal{S}_{|w|}} \mathbb{P}(S \mid w) \cdot \mathrm{DPdist}(S, S^*)$$

$$= |w| - \sum_{(i,j)\in S} 2p_w(i,j) - \sum_{i \text{ unpaired in } S} q_i \in \mathbb{R}_+$$

where $p_w(i,j)$ is the base pair probability of (i,j) and q_i is the probability of being unpaired for nucleotide i.

The use of ensemble defect requires the target structure to be the representative structure in the ensemble induced by the design w.

3.1.3 *Constraints*

Additional constraints are also considered in RNA design, such as imposing or forbidding a certain sequence pattern in the design or aiming a specific GC content for designed sequences. More formally, these goals can be seen as a set of constraints imposed on sequences $\mathcal{C} = \{c_1, \ldots, c_l\}$. Each constraint $c_i : \Sigma^* \to \{\mathsf{True}, \mathsf{False}\}$ is a function returning a boolean given a sequence. In general, constraint is defined on partial sequence, *i.e.*, the content of a selected subset of positions. Then, the sequence space for both positive and negative design is limited from Σ^* to $\mathcal{A}_\mathcal{C}$ the set of sequences compatible with $\mathcal{C}$, $\mathcal{A}_\mathcal{C} := \{w \in \Sigma^*; \forall c \in \mathcal{C}, c(w) \text{ is } \mathsf{True}\}$. For example, imposing a known aptamer sequence for better binding affinity [23].

3.2 STATE OF THE ART

Given a secondary structure S^* of length n with k base pairs. The amount of RNA sequences compatible with S^* is up to $6^k 4^{n-2k}$, *i.e.*, exponentially growing on the target length. It is unrealistic to find design sequences that adopt the target as MFE structure and satisfy defect conditions with a simple brute force method. Moreover, the classic inverse folding problem is shown to be NP-complete in the Nussinov-Jacobson model [7]. Several heuristic methods have been proposed to work around the hardness. Some approaches follow a similar workflow: an initial seed sequence generation for positive design followed by a local search optimization for negative design purposes.

3.2.1 *RNAinverse*

RNAinverse [41] is the first negative design approach for the inverse folding problem. Let S^* be the secondary target structure. Starting from an initial compatible sequence w_0, RNAinverse performs a random work starting in sequence space with optimization toward the target. At step i, a sequence w is obtained from the previous one with a mutation on an unpaired nucleotide or a base pair. The sequence w is accepted if the distance of the MFE structure to the target decreases,

$$\mathrm{dist}(\mathsf{MFE}(w), S^*) < \mathrm{dist}(\mathsf{MFE}(w_i), S^*) \implies w_{i+1} = w.$$

The walk stops if a design is found. When there is no more available mutation to improve the distance to the target, *i.e.*, local minima, the local optimization restarts with a new initial sequence.

To reduce the computation time on evaluating sequence, RNAinverse determines the best sequence by block. Considering structure loop decomposition as a tree, in which a node is a loop, the approach travels the tree in post-order to complete the sequence. Starting with a hairpin, RNAinverse looks for the sequence that folds into the current target as the MFE structure, then adds another loop into the current target. One can also consider the probability defect as the design objective of RNAinverse, despite the increase of time computation due to the defect evaluation on the entire sequence.

3.2.2 *Optimization for negative design*

As a pioneer, RNAinverse established a workflow followed by its successors for the RNA negative design problem: explore the sequence space with optimization. As mentioned above, RNAinverse performs local optimization using random walk starting from a seed sequence. However, local minima problem is sometimes observed during the random walk. INFO-RNA [12] overcomes the issue using stochastic local search, which accepts a worse sequence with a fixed probability. As for the design objective, NUPACK [96] targets the optimal ensemble defect in the optimization step.

Other optimization approaches are also proposed for the RNA design problem:

- AntaRNA [47] uses a nature-inspired ant colony optimization algorithm. Ants walk through a decision tree encoding the whole sequence space. The approach returns the sequences with a higher score while controlling GC content.
- MODENA [82] considers sequence affinity and specificity as two objective functions. It uses the Multi-objective optimization approach to search weak Pareto optimal design solutions.
- RNAifold uses constraint programming to determine the optimal design sequence [32]. It explores a larger sequence space with the integration of large neighborhood search.
- DSS-OPT [59] adopts simulated annealing to optimise design sequence within the sequence space using the Newtonian dynamics.
- MCTS-RNA [95] uses a heuristic sequence sampling algorithm with Monte Carlo Tree Search, a strategy initially developed for GO gaming [15].

3.2.3 *Generating compatible sequences for target structure(s)*

Andronescu *et al.* [2] showed that starting with initial sequences with a higher affinity to the target structure, *i.e.*, positive design objective, achieves a better performance after local search. Given a target structure, as implemented in RNA-SSD, the approach

initializes the seed sequence with a different nucleotide probability toward a low free-energy. It favors CG in helix and AU for unpaired nucleotides as (C, G) base pair stack is energetically more stable. INFO-RNA [12] uses the Dynamic Programming (DP) algorithm to find initial sequence in the sequence space with the minimum free-energy for the target structure using the Turner energy model. A global sequence sampling based on Boltzmann-weighted distribution is achieved by IncaRNAtion [69]. The approach considers a simplified energy model assuming that structure energy is an additive contribution of base pairs stacks. Sequences are sampled with stochastic backtracking after dual partition function computation from DP scheme.

Furthermore, IncaRNAtion adopts the strategy of multi-dimensional Boltzmann sampling [5] to controls sequence GC content as a constrained design objective. Zhou *et al.* [98] integrated automata in the framework to sample sequences for a given sequence pattern. For multiple hard constraints given, *i.e.*, multiple target structures, the complexity of finding compatible sequences increases as a nucleotide can have more than one parter. It is then insufficient to assign paired nucleotides to base pair.

One can represent the hard constraints introduced by target structures as a dependency graph, in which vertices are nucleotides, and two vertices are connected if they form a base pair in one of the target structures. When only two target structures are concerned, the degree of a vertex is at most 2, meaning that each connected component in the graph is either a path or a cycle. A compatible sequence is easily obtained by alternatively assigning C and G in each connected component. Flamm *et al.* [31] developed switch.pl to perform a uniform sequence sampling for two target structures. When there are more than two target structures, RNAblueprint [35] uniformly samples compatible sequences using graph coloring strategy, assuming each nucleotide is a color. The approach precomputes the number of compatible sequences on each subgraph in a decomposition of the dependency graph. Counting compatible sequences for multiple targets is later shown to be #P-hard problem [37]. A Fixed-Parameter Trackable (FPT) algorithm is proposed, as implemented in RNARedPrint [37], to sample sequences based on each target energy with a precomputation of dual partition function using DP algorithm. Using the same strategy as IncaRNAtion, RNARedPrint controls sequence GC content and targets specific target energies in sampled sequences. The algorithmic detail is presented later in Chapter 8 while introducing an extension of the RNARedPrint framework.

4 ANALYTIC COMBINATORICS

Analytic Combinatorics is a branch of discrete applied Mathematics combining combinatorics and complex analysis [30]. In enumerative combinatorics, the goal is to study an object, or a family of objects, through is quantitative properties using a variety of tools, including decompositions, bijections, formal languages, and generating functions. The latter is treated as a function in complex analysis, which gives a different sight for the object of interest, such as property statistics. This chapter presents examples using RNA secondary structure, definitions and results in analytic combinatorics that are necessary to the exposition of the first part of this thesis.

4.1 FORMAL LANGUAGE

Definition 4.1 (Alphabet): An *alphabet* Σ is a set with more than one element, which is called a *letter* or a *symbol*.

A word over an alphabet Σ is a sequence of letters in Σ. The set of all words over Σ, denoted by Σ^*, is the free monoid over Σ with the empty word ε as the identity element and string concatenation, denoted by $\cdot$, as product operation such that $a, b \in \Sigma^* \implies a \cdot b \in \Sigma^*$.

Definition 4.2 (Language): A *language* $\mathcal{L}$ over an alphabet Σ is a subset of Σ^*, $\mathcal{L} \subseteq \Sigma^*$.

Example (RNA sequence and Motzkin word):

- *An RNA sequence is a word over the alphabet* $\Sigma_w = \{A, C, G, U\}$ *in the context of formal language.*
- *A Motzkin word is, similar to Dyck word, a well-parenthesized expression composed of letters in alphabet* $\Sigma_S = \{(,), \bullet\}$.

Language is usually defined using a grammar, which contains a set of production rules $\alpha \to \beta$. The left-hand side α and the right-hand side β are sequences composed of an alphabet Σ and a set of nonterminal symbols $\mathcal{N}$, *i.e.*, $\alpha, \beta \in (\mathcal{N} \cup \Sigma)^*$. In a word w, a nonterminal symbol points out the location where production rules can be applied, *i.e*, w is not completed yet. A word consists only of letters in Σ is completed, on which no production rule can be applied. In the Chomsky hierarchy, grammar is classified depending on the rule type:

- Type-0, Recursively enumerable $\gamma \to \alpha$. No restriction is applied on both sides;
- Type-1, Context-sensitive $\alpha S \beta \to \alpha \gamma \beta$. Nonterminal symbol S is substituted by γ depending on the context upstream α and downstream β;
- Type-2, Context-free $S \to \alpha$. The left-hand side is restricted to a nonterminal symbol;
- Type-3, Regular $S \to a$ or $S \to aT$. Only one nonterminal symbol is allowed on the left-hand side, while only a letter potentially followed by a nonterminal symbol is allowed on the right-hand side.

where S is a nonterminal symbol, a is a letter, and α, β, γ are sequences in $(\mathcal{N} \cup \Sigma)^*$ such that γ is not empty. In this thesis, we only consider the class of grammars using type-2 production rules, Context-Free Grammar.

Definition 4.3 (Context-Free Grammar): A *Context-Free Grammar (CFG)* is given by $G = (\Sigma, \mathcal{N}, S_0, \mathcal{R})$ such that

- Σ is an alphabet, the element is named terminal symbol;
- $\mathcal{N}$ is a finite set of nonterminal symbols;
- $S_0 \in \mathcal{N}$ is the start symbol;
- $\mathcal{R} \subset \mathcal{N} \times (\mathcal{N} \cup \Sigma)^*$ is a finite set of production rules.

A rule $(S, w) \in \mathcal{R}$ is usually denoted by $S \to w$. Rules having the same nonterminal symbol on the left-hand side, $S \to w_1, \ldots, S \to w_n$, are abbreviated by $S \to w_1 \mid \cdots \mid w_n$.

Definition 4.4 (Direct derivation): Let $G = (\Sigma, \mathcal{N}, S_0, \mathcal{R})$ be a grammar and u, v be two words of $(\mathcal{N} \cup \Sigma)^*$. We say v is directly derived from u, denoted by $u \to_G v$, if there exists a production rule $(\alpha, \beta) \in \mathcal{R}$ and two words w_1, w_2 of $(\mathcal{N} \cup \Sigma)^*$ such that

$$u = w_1 \cdot \alpha \cdot w_2 \qquad \text{and} \qquad v = w_1 \cdot \beta \cdot w_2.$$

Definition 4.5 (Derivation): Let $G = (\Sigma, \mathcal{N}, S_0, \mathcal{R})$ be a grammar and w a word over Σ. We say a word w is derived from a nonterminal symbol $S \in \mathcal{N}$, denoted by $S \rightsquigarrow_G w$, if there exists a finite sequence of words $w_1, \ldots, w_k \in (\mathcal{N} \cup \Sigma)^*$ with $k \in \mathbb{N}^*$ such that

$$S \to_G w_1 \to_G \cdots \to_G w_k \to_G w.$$

A *leftmost* derivation is a derivation, where production rule is applied on the leftmost nonterminal symbol every time.

Definition 4.6 (Language of a grammar): The language $\mathcal{L}_{G,S}$ generated from a nonterminal symbol S in grammar G is the set of words derived from S,

$$\mathcal{L}_{G,S} := \{w \in \Sigma^*;\ S \rightsquigarrow_G w\}.$$

The language $\mathcal{L}_G$ generated from a grammar G is the language generated from the start symbol S_0,

$$\mathcal{L}_G := \mathcal{L}_{G,S_0} = \{w \in \Sigma^*;\ S_0 \rightsquigarrow_G w\}.$$

The index G is usually omitted when there is no ambiguity.

Example (Grammar for Motzkin words): *Reminder that a Motzkin word is a well-parenthesized expression over the alphabet $\Sigma_S = \{(,), \bullet\}$. Motzkin words can be generated from the grammar $G_S^{bis} = (\Sigma_S, \{S\}, S, \mathcal{R})$ with the production rules*

$$\begin{aligned}
&R_1 : S \to \bullet S\\
&R_2 : S \to S\bullet\\
&R_3 : S \to (S)S\\
&R_4 : S \to \varepsilon
\end{aligned}$$

For example, the word $\bullet()\bullet$ is derived from the start symbol via the following leftmost derivation

$$S \xrightarrow{R_1} \bullet S \xrightarrow{R_2} \bullet S\bullet \xrightarrow{R_3} \bullet(S)S\bullet \xrightarrow{R_4} \bullet(\varepsilon)S\bullet \xrightarrow{R_4} \bullet(\varepsilon)\varepsilon\bullet = \bullet()\bullet.$$

One may observe that the leftmost derivation is not unique, the following is another one for $\bullet()\bullet$ by switching the first two rules,

$$S \xrightarrow{R_2} S\bullet \xrightarrow{R_1} \bullet S\bullet \xrightarrow{R_3} \bullet(S)S\bullet \xrightarrow{R_4} \bullet(\varepsilon)S\bullet \xrightarrow{R_4} \bullet(\varepsilon)\varepsilon\bullet = \bullet()\bullet.$$

Definition 4.7 (Unambiguous Grammar): A grammar G is *unambiguous* if all words $w \in \mathcal{L}_G$ generated from grammar have a unique leftmost derivation.

Example (Unambiguous grammar for Motzkin words): *One of the common used unambiguous grammars for Motzkin words is $G_S^0 = (\Sigma_S, \{S_0\}, S_0, \mathcal{R})$ with the production rules*

$$S_0 \to (S_0)S_0 \mid \bullet S_0 \mid \varepsilon.$$

4.2 ANALYTIC COMBINATORICS

Definition 4.8 (Combinatorial Class): A *combinatorial class* $\mathcal{A}$ is a set associated with a size function $|\cdot| : \mathcal{A} \to \mathbb{N}$ such that the subset, denoted by $\mathcal{A}_n \subset \mathcal{A}$ of elements of any given size n is finite.

Example:

- *Language $\mathcal{L}_G$ derived from a grammar G is naturally a combinatorial class with the size is the length of word.*
- *Secondary structures can be seen as a combinatorial class with the associated size function returns the amount of nucleotides given a secondary structure.*

4.2.1 *Ordinary Generating Function*

Definition 4.9 (Ordinary Generating Function): Let $\mathcal{A}$ be a combinatorial class, the Ordinary Generating Function (OGF) of $\mathcal{A}$ is the power series

$$A(z) = \sum_{a \in \mathcal{A}} z^{|a|} = \sum_n a_n z^n$$

where $a_n := |\mathcal{A}_n|$ is the number of elements of size n in $\mathcal{A}$.

The use of OGF allows us to consider some basic operations on combinatorial classes. Let $\mathcal{A}, \mathcal{B}, \mathcal{C}$ be three combinatorial classes and, for any positive integer n, $a_n := |\mathcal{A}_n|, b_n := |\mathcal{B}_n|, c_n := |\mathcal{C}_n|$ be the cardinality of subsets restricted to size n.

- If $\mathcal{A} = \mathcal{B} \cup \mathcal{C}$, one has $a_n = b_n + c_n$ if sets $\mathcal{B}_n$ and $\mathcal{C}_n$ are disjoint for any n;
- Let $\times$ denote the Cartesian product. If $\mathcal{A} = \mathcal{B} \times \mathcal{C} = \{b \cdot c;\ b \in \mathcal{B}, c \in \mathcal{C}\}$ and the size function is additive upon concatenating two elements, one has, for any n,

$$a_n = \sum_{i=0}^{n} b_i \cdot c_{n-i} \qquad \text{since} \qquad \mathcal{A}_n = \bigcup_{i=0}^{n} \{b \cdot c;\ b \in \mathcal{B}_i, c \in \mathcal{C}_{n-i}\}.$$

Proposition 4.1: *Let $\mathcal{A}, \mathcal{B}, \mathcal{C}$ be three combinatorial classes and $A(z), B(z), C(z)$ be the OGF for each.*

- *If $\mathcal{A} = \mathcal{B} \cup \mathcal{C}$ and $\mathcal{B} \cap \mathcal{C} = \varnothing$, then $A(z) = B(z) + C(z)$;*
- *If $\mathcal{A} = \mathcal{B} \times \mathcal{C}$, then $A(z) = B(z) \cdot C(z)$.*

FROM GRAMMAR TO GENERATING FUNCTION The Dyck-Schützenberger-Viennot (DSV) method [10, 53], sometimes referred to as the symbolic method in the subsequent work of Flajolet and colleagues, is often used to find the OGF for a combinatorial class $\mathcal{A}$. The method consists of three steps, describing the class with grammar, transforming the production rules into a system of functional equations, and resolving the system to get the OGF.

1. **Describe combinatorial class.** Finding an unambiguous grammar forming language $\mathcal{L}$ such that, for any size n, the set $\mathcal{L}_n$ has a bijection with the combinatorial class $\mathcal{A}_n$. It requires to prove

   - **Completeness.** Any object in $\mathcal{A}_n$ matches to at least one word of length n derived from the grammar;
   - **Correctness.** Any word derived from the grammar corresponds to an object in $\mathcal{A}_n$.

   With unambiguity, this shows a bijection between the words constructed by the grammar and the objects of the combinatorial class of interest.

2. **Construct a system of functional equations.** Next, we transform the production rules into a system of functional equations. Each nonterminal symbol $S \in \mathcal{N}$ is transformed into $S(z)$, the OGF of the language $\mathcal{L}_S$ derived from S, while each terminal symbol is transformed into z. Accordingly, the language $\mathcal{L}_a$ of a letter $a \in \Sigma$ consists of a itself, $\mathcal{L}_a = \{a\}$. Thus, the associated OGF is equal to $z^{|a|} = z$.

   Given a production rule $S \to \beta$ with $\beta \in (\mathcal{N} \cup \Sigma)^*$, the transformation to an equation of OGF is a direct application of Proposition 4.1. Let $\beta = x_1 \cdots x_l$ with $x_i \in \mathcal{N} \cup \Sigma$ is a (non)terminal symbol. Then, a word derived from S is a word in $\mathcal{L}_{x_1} \times \cdots \times \mathcal{L}_{x_l}$, which gives the equation below

   $$S(z) = \prod_{x \in \beta} \begin{cases} 1 & \text{if } x = \varepsilon \\ z & \text{if } x = a \in \Sigma \\ B(z) & \text{if } x = B \in \mathcal{N}. \end{cases} \tag{4.1}$$

   For abbreviated production rules $S \to \beta \mid \gamma$, the language derived from S is the union of languages derived from β and γ. Let $\beta(z)$ (resp. $\gamma(z)$) be the OGF of the language derived from β (resp. γ), computed using Equation 4.1. One has $S(z) = \beta(z) + \gamma(z)$. Note that the OGF $S_0(z)$ associated to the start symbol S_0 is the OGF of the combinatorial class of interest since the language derived from grammar forms a bijection.

3. **Solve the system of functional equations.** It is possible to eliminate the OGF for nonterminal symbols other than the start symbol S_0 from the system and compute an expression for $S_0(z)$, OGF of the language derived from S_0, *i.e.*, grammar. The complexity of resolving the system is determined by the degree of the OGF in the system.

Example (RNA secondary structure): *We apply the* DSV *method on secondary structure with minimum distance* $\theta = 0$.

- *With* $()$ *representing base pair and* $\bullet$ *standing for unpaired base, it is easy to see a bijection between the set of secondary structures and the language derived from the grammar* G_S^0 *described in the previous example.*
- *Let* $S_0(z)$ *be the* OGF *for the start symbol* S_0*. The transformation for the first rule* $S_0 \to (S_0)S_0$ *is*

$$S_0(z) = \psi((S_0)S_0) = z\psi(S_0)S_0) = zS_0(z)\Psi()S_0) = z^2S_0(z)\psi(S_0) = z^2S_0(z)^2.$$

  Same analogy for the entire production rules $S_0 \to (S_0)S_0 \mid \bullet S_0 \mid \varepsilon$ *yields a quadratic equation of* $S_0(z)$,

$$S_0(z) = z^2S_0(z)^2 + zS_0(z) + 1.$$

- *Resolving the equation gives two functions of* z,

$$S^+(z) = \frac{1 - z + \sqrt{(1+z)(1-3z)}}{2z^2} \quad \textit{and} \quad S^-(z) = \frac{1 - z - \sqrt{(1+z)(1-3z)}}{2z^2}.$$

  We will see later that $S_0(z) = S^-(z)$.

4.2.2 *Asymptotic value for OGF coefficients*

In this section, Ordinary Generating Function (OGF) is seen as a function of z as found in the previous example.

Definition 4.10 (Coefficient of Generating Function): Let $f(z)$ be the OGF of a combinatorial class $\mathcal{A}$. We use $[z^n]\, f(z)$ to denote the coefficient of z^n in $f(z)$, *i.e.*, $[z^n]\, f(z) = |\mathcal{A}_n|$.

It is, in general, hard to find an explicit expression for $[z^n]\, f(z)$. An alternative way is to compute the asymptotic value with the help of singularity analysis [30], which is summarized by two principles,

- **First Principle of Coefficient Asymptotics**. The location of singularities dictates the exponential growth α^n of coefficients, where α is the inverse of the dominant singularity;
- **Second Principle of Coefficient Asymptotics**. The nature of singularities determines the associate subexponential factor $\vartheta(n)$, $\lim_{n\to+\infty} \vartheta(n)^{1/n} = 1$.

These two suggests that the asymptotic value of coefficient $[z^n]\, f(z)$ is in the form of $\alpha^n\vartheta(n)$. Singularities of $f(z)$ are the points in the complex plane where $f(z)$ ceases to be analytic. Pringsheim's Theorem limits the singularities to be in real numbers $\mathbb{R}$ for OGF.

Theorem 4.2 (Pringsheim's Theorem): *If $f(z)$ is representable at the origin by a series expansion that has non-negative coefficients and radius of convergence $R \in \mathbb{R}$, then the point $z = R$ is a singularity of $f(z)$.*

The dominant singularity ρ of $f(z)$ is the smallest non-zero singularity of $f(z)$ in absolute value,

$$\rho := \sup\{r > 0;\ f \text{ analytic at all points of } 0 \leqslant z < r\}.$$

Theorem 4.3 (Exponential Growth Formula): *Let $\rho \in \mathbb{R}_+$ be the dominant singularity (in absolute value) of an Ordinary Generating Function $f(z)$. One has*

$$\lim_{n \to +\infty} [z^n]\, f(z) = \left(\frac{1}{\rho}\right)^n.$$

The formula is referred to the First Principle of Coefficient Asymptotics.

The Second Principle of Coefficient Asymptotics is explained in Flajolet and Odlyzko Theorem [29].

Theorem 4.4 (Flajolet and Odlyzko Theorem): *Let α be a non-integer number, ρ be a non-zero number and let $f(z)$ and $g(z)$ be two functions such that*

$$f(z) = (1 - z/\rho)^{-\alpha} \qquad \textit{and} \qquad g(z) = o((1 - z/\rho)^{-\alpha}).$$

The coefficient of z^n in $f(z)$ and $g(z)$ are

$$[z^n]\, f(z) = \frac{\rho^{-n} n^{\alpha-1}}{\Gamma(\alpha)} \left(1 + \frac{\alpha(\alpha-1)}{2n} + o(\frac{1}{n})\right)$$

$$[z^n]\, g(z) = o(\rho^{-n} n^{\alpha-1})$$

where Γ is the Gamma function, $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$.

PROCESS OF SINGULARITY ANALYSIS Let $f(z)$ be an OGF. The process of singularity analysis to determine the asymptotic coefficient is as follows,

1. Determine the dominant singularity of $f(z)$;
2. Expand $f(z)$ around the dominant singular;
3. Apply Flajolet and Odlyzko Theorem on each term in the expansion to obtain asymptotic equivalent of $[z^n]\, f(z)$.

Example (Asymptotic number for secondary structures): *From previous example, the* OGF *of secondary structures with $\theta = 0$ is either $S^+(z)$ or $S^-(z)$*

$$S^+(z) = \frac{1 - z + \sqrt{(1+z)(1-3z)}}{2z^2} \quad \text{and} \quad S^-(z) = \frac{1 - z - \sqrt{(1+z)(1-3z)}}{2z^2}.$$

1. *Both solutions have the dominant singularity $\rho = 1/3$ since the value in the square root cannot be negative;*
2. *Singularity expansion of $S^-(z)$ on $z = \rho = 1/3$ gives*

$$S^-(z) = 3 - 3\sqrt{3}(1 - \frac{z}{\rho})^{\frac{1}{2}} + o((1 - \frac{z}{\rho})^{\frac{1}{2}});$$

3. *Applying Flajolet and Odlyzko Theorem with $\alpha = -1/2$ and $\Gamma(-1/2) = -2\sqrt{\pi}$, we obtain*

$$[z^n]\, S^-(z) = \frac{3\sqrt{3}}{2\sqrt{\pi}}(1 + o(1)) \times 3^n n^{-\frac{3}{2}} + o(3^n n^{-\frac{3}{2}}).$$

Using the same process for $S^+(z)$, the value of $[z^n]\, S^+(z)$ is negative when n is odd which against the non-negative number of secondary structures. Thus, $S^-(z)$ is the only possible OGF *for secondary structures and the asymptotic amount for structures of length n is*

$$[z^n]\, S_0(z) = \frac{3\sqrt{3}}{2\sqrt{\pi}} \times 3^n n^{-\frac{3}{2}} + o(3^n n^{-\frac{3}{2}}).$$

In the special case, where OGF is expressed in a recursive form $f(z) = G(z, f(z))$, one can use Bender-Meir-Moon Theorem [61] to compute the subexponential factor. The theorem is first proposed by Bender in 1974, then renewed by Meir and Moon.

Theorem 4.5 (Bender-Meir-Moon Theorem): *Suppose that the generating function $f(z)$ is analytic at $z = 0$, that $f_n \geqslant 0$ for all n, and that $f(z) = G(z, f(z))$, where $G(z, y) = \sum_{m,n \geqslant 0} g_{m,n} z^m y^n$ such that there exists three positive real numbers δ, r, s satisfying*

- *$G(z, y)$ is analytic in $|z| < r + \delta$ and $|y| < s + \delta$*
- *$G(r, s) = s$, $\frac{\partial}{\partial y} G(r, s) = 1$*
- *$G_z(r, s) := \frac{\partial}{\partial z} G(r, s) \neq 0$ and $G_{y,y} := \frac{\partial^2}{\partial y^2} G \neq 0$*

If $g_{m,n}$ is non-negative real number for all m, n, $g_{0,0} = 0$, $g_{0,1} \neq 1$, and $g_{m,n} > 0$ for some m and for some $n \geqslant 2$, then

$$f_n = [z^n] f(z) \sim \sqrt{\frac{r G_z(r, s)}{2\pi G_{y,y}(r, s)}}\, r^{-n} n^{-\frac{3}{2}}.$$

Example (Asymptotic number for secondary structures 2 [77]): *Reminder that the* OGF $S_0(z)$ *of secondary structures with minimum distance* $\theta = 0$ *satisfies*

$$S_0(z) = z^2 S_0(z)^2 + zS_0(z) + 1.$$

In other words, $S_0(z)$ *is in a recursive form* $S_0(z) = G(z, y)$ *with* $G(z, y) = 1 + zy + z^2y^2$. *It is easy to verify that all conditions for Bender-Meir-Moon Theorem are fulfilled with* $r = 1/3$, $s = 3$, *and* $\delta = 1/3$. *Computing the partial derivative of* G *gives* $G_z(r, s) = 9$ *and* $G_{y,y}(r, s) = 2/9$. *Thus, the asymptotic amount for secondary structures of length* n *is*

$$[z^n]\, S_0(z) \sim \sqrt{\frac{\frac{1}{3} \cdot 9}{2\pi \cdot \frac{2}{9}}} \times 3^n n^{-\frac{3}{2}} = \frac{3\sqrt{3}}{2\sqrt{\pi}} \times 3^n n^{-\frac{3}{2}}$$

which is exactly same as the one obtained in the previous example.

4.2.3 *Bivariate Generating Function*

Despite that the Ordinary Generating Function provides a good estimation for the growth of the combinatorial class in the function of the size, it is not enough to study others object properties in many cases, such as the expected value of a property in the ensemble. The object property introduces an additional parameter, characterized as a function, named feature, associating a value to each combinatorial object.

Definition 4.11 (Bivariate Generating Function): Let $\mathcal{A}$ be a combinatorial class and $F : \mathcal{A} \to \mathbb{R}$ be a feature. Bivariate Generating Function (BGF) $A(z, u)$ is defined as

$$A(z, u) = \sum_{a \in \mathcal{A}} z^{|a|} u^{F(a)} = \sum_{n \leqslant 0, k \leqslant 0} a_{n,k} z^n u^k.$$

where $a_{n,k}$ is the amount of objects of size n having value k for the feature F. We say the variable u marks the feature F.

In order to use the DSV method to construct a system of functional equations from a grammar, we consider a special type of feature of which the value is expressed as the additive sum over production rules.

Definition 4.12 (Additive feature): Let $\mathcal{A}$ be a combinatorial class and $G = (\Sigma, \mathcal{N}, S_0, \mathcal{R})$ be a grammar such that a bijection exists for the derived language $\mathcal{L}_G$ and $\mathcal{A}$. A feature F is additive if there exists a function $f : \mathcal{R} \to \mathbb{R}$ defined on production rules such that for an object $a \in \mathcal{A}$ and its mapped word $w \in \mathcal{L}_G$,

$$F(a) = \sum_{r \in p_w} f(r)$$

where p_w is the set of rules applied in the leftmost derivation $S \rightsquigarrow w$ of w.

Similar to Equation 4.1, the system of functional equation is constructed by transforming each production rule $r : A \to \beta$ to an equation of BGF while introducing variable u to mark feature value $f(r)$,

$$A(z,u) = u^{f(r)} \times \beta(z) = u^{f(r)} \times \prod_{x \in \beta} \begin{cases} 1 & \text{if } x = \varepsilon \\ z & \text{if } x \in \Sigma \\ B(z,u) & \text{if } x = B \in \mathcal{N} \end{cases}$$

where $\beta(z)$ is the OGF of the set of words derived from β.

Example (Base pairs in secondary structure): *Since base pair stabilizes RNA structure, it is natural to ask the base pair number in a secondary structure. Secondary structure with the minimum distance $\theta = 0$ is generated with the rules*

$$S_0 \to (S_0)S_0 \mid \bullet S_0 \mid \varepsilon.$$

Only the first rule $S_0 \to (S_0)S_0$ involves one base pair. The value of feature f is then 0 for the first rule and 0 for the others. Let $S_0(z,u)$ be the BGF with z and u mark, respectively, the size and the amount of base pairs. We obtain from production rules,

$$S_0(z,u) = z^2 u S_0(z,u)^2 + z S_0(z,u) + 1.$$

Thus,

$$S_0(z,u) = \frac{1 - z - \sqrt{(z-1)^2 - 4z^2u}}{2z^2u}.$$

Similar result has been shown on a more realistic secondary structure set with Knudsen-Hein grammar [48, 67].

Let X_n be a random variable for the feature of object of size n. One can compute the expected feature value $\mathbb{E}[X_n = k]$ from the BGF,

$$\mathbb{E}[X_n = k] = \frac{\sum_{a \in \mathcal{A}_n} F(a)}{|\mathcal{A}_n|} = \frac{[z^n] \sum_{a \in \mathcal{A}_n} F(a) u^{F(a)-1} z^n \big|_{u=1}}{[z^n] \sum_{a \in \mathcal{A}_n} z^n}.$$

Thus,

$$\mathbb{E}[X_n] = \frac{[z^n] \left.\frac{\partial A(z,u)}{\partial u}\right|_{u=1}}{[z^n]\, A(z,1)}.$$

The expected value is also called the first moment. One can obtain other moments similarly.

Proposition 4.6 (Factorial Moment from BGF): *The r-th factorial moment of X_n, within the uniform distribution over objects of size n, is computed from the BGF $A(z,u)$ by r-fold differentiation followed by evaluation at 1,*

$$\mathbb{E}[X(X-1)\cdots(X-r+1)] = \frac{[z^n]\ \frac{\partial^r A(z,u)}{\partial u}\Big|_{u=1}}{[z^n]\,A(z,1)}.$$

In particular, for the second moment,

$$\begin{aligned}\mathbb{E}[X^2] = \mathbb{E}[X] + \mathbb{E}[X(X-1)] &= \frac{[z^n]\ \frac{\partial A(z,u)}{\partial u}\Big|_{u=1}}{[z^n]\,A(z,1)} + \frac{[z^n]\ \frac{\partial^2 A(z,u)}{\partial u^2}\Big|_{u=1}}{[z^n]\,A(z,1)} \\ &= \frac{[z^n]\ \frac{\partial}{\partial u}\left(u\cdot\frac{\partial A(z,u)}{\partial u}\right)\Big|_{u=1}}{[z^n]\,A(z,1)}.\end{aligned}$$

The variance of feature value can then be computed with $\mathbb{V}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$.

Example (Expected base pair number in secondary structure): *The partial derivative of $S_0(z,u)$ with respect to u is*

$$\frac{\partial S_0(z,u)}{\partial u} = \frac{1}{u((1-z)^2-4z^2u)^{\frac{1}{2}}} + \frac{-1+z+((1-z)^2-4z^2u)^{\frac{1}{2}}}{2z^2u}.$$

With $u=1$,

$$\begin{aligned}\frac{\partial S_0(z,u)}{\partial u}\Big|_{u=1} &= \frac{1}{(1+z)^{\frac{1}{2}}(1-3z)^{\frac{1}{2}}} + \frac{1-z+(1+z)^{\frac{1}{2}}(1-3z)^{\frac{1}{2}}}{2z^2} \\ &= \frac{z-1}{2z^2} + \frac{1-2z-z^2}{2z^2(1+z)^{\frac{1}{2}}} \times (1-3z)^{-\frac{1}{2}}\end{aligned}$$

Expanding on the dominant singularity $z=\rho=1/3$ gives

$$\frac{\partial S_0(z,u)}{\partial u}\Big|_{u=1} = -3 + \frac{\sqrt{3}}{2}(1-\frac{z}{\rho})^{-\frac{1}{2}} + o((1-\frac{z}{\rho})^{-\frac{1}{2}}).$$

The asymptotic value for n-th coefficient is obtained by applying Flajolet and Odlyzko Theorem with $\alpha = 1/2$ and $\Gamma(1/2)=\sqrt{\pi}$,

$$[z^n]\ \frac{\partial S_0(z,u)}{\partial u}\Big|_{u=1} = \frac{\sqrt{3}}{2\sqrt{\pi}} \times 3^n n^{-\frac{1}{2}} + o(3^n n^{-\frac{1}{2}}).$$

Since $[z^n]\,S_0(z,1) = [z^n]\,S_0(z) = \frac{3\sqrt{3}}{2\sqrt{\pi}} \times 3^n n^{-\frac{3}{2}} + o(3^n n^{-\frac{3}{2}})$, taking the ratio gives the expected base pair number in a random secondary structure of length n is $\frac{n}{3} + \mathcal{O}(1)$. In other words, in average, two third of nucleotides are paired in a random structure.

As seen in the example, the expected feature value is linear to the object size. In fact, Drmota–Lalley–Woods Theorem [25, 50, 92] shows that the variance is also linear to the size and the feature value follows a Gaussian limiting distribution if the system of equations is strongly connected.

Theorem 4.7 (Drmota–Lalley–Woods Theorem): *Let $\mathcal{A}$ be a combinatorial class and G be a grammar for $\mathcal{A}$. Considering an additive feature F, the BGF for the feature is denoted by $A(z,u)$ and let X_n be the random variable for feature value of an object in $\mathcal{A}_n$. If the system of functional equations for $A(z,u)$ constructed from the grammar G, then X has a Gaussian limiting distribution with mean $\mathbb{E}[X_n]$ and variance $\mathbb{V}[X_n]$ linear to n,*

$$\mathbb{E}[X_n] = \mu n + \mathcal{O}(1) \qquad \text{and} \qquad \mathbb{V}[X_n] = \sigma^2 n + \mathcal{O}(1)$$

where μ and σ are two constants. In addition, let $\rho(u)$ be the dominant singularity of $A(z,u)$ in the function of u. Then,

$$\mu = -\frac{\rho'(1)}{\rho(1)} \qquad \text{and} \qquad \sigma^2 = -\frac{\rho''(1)}{\rho(1)} + \mu^2 + \mu.$$

Note that the theorem statement is adapted to the context of BGF and grammar.

Example (Expected base pair number in secondary structure 2): *Reminder that the system for the BGF $S_0(z,u)$ is*

$$S_0(z,u) = z^2 u S_0(z,u)^2 + z S_0(z,u) + 1$$

which is strongly connected since it consists of only one BGF. Furthermore, we have

$$S_0(z,u) = \frac{1 - z - \sqrt{(z-1)^2 - 4z^2u}}{2z^2u}$$

is not analytic when the value within the square root is negative. Thus, the dominant singularity $z = \rho(u)$ is a solution of $(z-1)^2 - 4z^2u = 0$. Solving the equation and using the fact that the dominant singularity is $1/3$ when $u = 0$ yield

$$\begin{aligned}
\rho(u) &= \frac{2\sqrt{u} - 1}{4u - 1} \\
\rho'(u) &= \frac{4 - u^{-\frac{1}{2}} - 4u^{\frac{1}{2}}}{(4u-1)^2} \\
\rho''(u) &= \frac{(\frac{1}{2}u^{-\frac{3}{2}} - 2u)(4u-1) - 8(4 - u^{-\frac{1}{2}} - 4u^{\frac{1}{2}})}{(4u-1)^3}
\end{aligned}$$

and $\rho(1) = 1/3, \rho'(1) = -1/9, \rho''(1) = 7/54$.
In conclusion, the amount of base pairs in a random secondary structure of length n follows a Gaussian limiting distribution $\mathcal{N}(1/3 \cdot n, 1/\sqrt{18} \cdot \sqrt{n})$.

Part II

UPPER-BOUND FOR DESIGNABLE RNA 2D STRUCTURES

5 UNDESIGNABLE MOTIFS

Haleš *et al.* [34] observed that if a secondary structure contains a multi-loop consisting of more than three open base pairs or a multi-loop having at least one unpaired position (Figure 5.1a), then the secondary structure is undesignable in the simple base pair based energy models. Similarly, Aguirre-Hernández, Hoos, and Condon [1] showed two motifs composed of two adjacent internal loops (Figure 5.1b) are undesignable in the usual Turner energy model. Forming a larger internal loop is thermodynamically more stable for any RNA sequence than folding into these two undesignable motifs.

This chapter aims to formalize and identify undesignable motifs in the RNA design problem with Turner energy model. We will start by introducing several definitions related to motif.

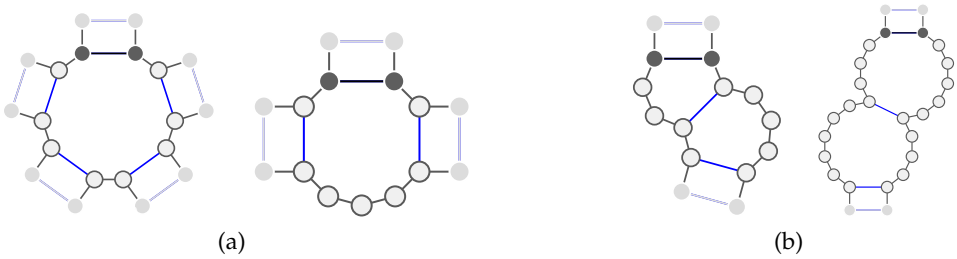


Figure 5.1: Examples of undesignable motifs in base pair based energy model [34] (a) and Turner energy model [1] (b).

5.1 MOTIF DEFINITION

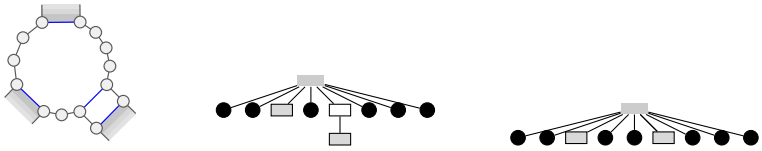
5.1.1 *Basic definition*

Definition 5.1 (Motif): A *motif* m is a rooted ordered tree such that the root and nodes are base pairs, and a leaf represents either an unpaired base or an open base pair, named *open paired leaf*. Its length, denoted by $|m|$, is the number of bases involved, and the number of open paired leaves is denoted by δ_m.

We use $\mathcal{M}$ to denote the set of motifs and $\mathcal{M}_n$ for the subset where motif length is restricted to n. Figure 5.2 shows an example of a motif. One can see a motif as

an object composed of several adjacent loops, each corresponding to a node and its children in the tree. In addition, one can notice that a primary loop is also a motif, called a *shallow motif*.

Definition 5.2 (Shallow motif): A shallow motif is a motif of height 1.



(a) (b) (c)

Figure 5.2: A motif m of length 14 with 2 open paired leaves presented as a graph (a) and as a tree (b). The motif is composed of a multi-loop and a stacking. (c) The shallow motif $\psi(m)$ returned by shallow operation.

Next, we define an operation ψ, named *reduction operation*, which returns a unique shallow motif $\psi(m)$ given a motif m. We show, in Figure 5.2c, the shallow motif returned while applying the reduction operation on the motif in Figure 5.2b.

Definition 5.3 (Reduction operation): Taken a motif as input, the reduction operation ψ duplicates and returns motif m while removing all base pairs in the motif that is not the root or the open paired leaves.

The obtained shallow motif has an equal length and the same open paired leaves amount as the given motif m, $|m| = |\psi(m)|$ and $\delta_m = \delta_{\psi(m)}$. In addition, the positions of open paired leaves remain the same in both motifs. Furthermore, based on reduction operation, we can define an equivalence relation for motifs. Given two motifs x and y, we say x is equivalent to y, denoted by $x \sim y$, if $\psi(x) = \psi(y)$. We then obtain a partition for motifs, in which each subset (or equivalence class) is represented by one shallow motif. Given a motif m, we use $[m] := \{x \in \mathcal{M}_{|m|};\ \psi(x) = \psi(x)\}$ to denote its equivalence class.

Similarly as for a secondary structure, a motif m can also be written in the dot-bracket notation. We use $(*)$ to represent an open paired leaf. In addition, we use m' to denote the partial motif delimited by the motif root, *i.e.*, $m = (m')$. It is easy to modify the classic secondary structure grammar into a grammar that builds the set of motifs with a given minimum distance.

Proposition 5.1 (Motif grammar): *Grammar below, starting with non-terminal symbol* M*, generates all motifs with a minimum distance* $\theta = 3$*,*

$$
\begin{aligned}
M &\to (U_3) \\
T &\to (U_3)T + (*)T + \bullet T + \varepsilon \\
U_3 &\to (U_3)T + (*)T + \bullet U_2 \\
U_2 &\to (U_3)T + (*)T + \bullet U_1 \\
U_1 &\to (U_3)T + (*)T + \bullet T
\end{aligned}
$$

Here, the non-terminal symbol U_i records the amount of bases needed to fulfill the minimum distance. The symbol $*$ is a letter of zero length, which represents any valid structure that can be placed within the open base pair while extending the motif.

Definition 5.4 (Motif extension): Given a motif m and a δ_m-tuple $(t_1, \cdots, t_{\delta_m})$ of motifs (resp. structures), we define $m \circ (t_1, \cdots, t_{\delta_m})$ the motif (resp. structure) obtained by making the i-th open paired leaf of m the parent of t_i for all i from 1 to δ_m.

In other words, we replace the i-th $*$ by t_i in the dot-bracket notation. We can then define motif occurrence used in Steyaert and Flajolet work [78].

Definition 5.5 (Motif occurrence):

- *Root occurrence*. A motif m is said to occur at the root of a secondary structure S^* (resp. a motif m^*) if there exists a δ_m-tuple of secondary structures (resp. motifs) $t_1, \cdots, t_{\delta_m}$ such that S^* (resp. m^*) is $m \circ (t_1, \cdots, t_{\delta_m})$.
- *Occurrence*. A motif m is said to occur in a secondary structure S^* (resp. motif m^*), denoted by $m \in S^*$ (resp. $m \in m^*$), if m occurs at the root of a subtree of S^* (resp. m^*).

Given a motif m and a secondary structure S (or a motif), one can find all occurrences of m in S in $\Theta(|m| \times |S|)$ using a simple pattern matching recursive algorithm as detailed in Algorithm 5.1.

As for string objects, two strings overlap if a suffix of one is a prefix of another, one can introduce a similar concept of overlapping to the motif. Analogously, a suffix of a string corresponds to a subtree of a motif, and a prefix is a smaller motif that occurs at the root. However, unlike a string object, we need to take structures extending from the motif into account because of open-paired leaves. Let $\mathcal{R}_m$ be the set of secondary structures featuring a root occurrence of motif m,

$$\mathcal{R}_m = \{m \circ (S_1, \ldots, S_{\delta_m}); (S_1, \ldots, S_{\delta_m}) \in \mathcal{S}^{\delta_m}\}.$$

Algorithm 5.1: Find motif occurrences in a secondary structure

```
Input  : A motif m and a secondary structure S
Output: 𝒪 a, possible empty, set of occurrence positions of m in S
Function FindAllOccurrences(m, S):
    𝒪 ← ∅;
    forall subtree t of S do
        if OccursAtRoot (m,t) then
            𝒪 ← 𝒪 ∪ {IndexOf(t)};
    return 𝒪
Function OccursAtRoot(m, t):
    r_m ← root of m;
    r_t ← root of t;
    if r_m = r_t = () then
        C_m := (c_m1, ··· , c_mk) ← children of r_m;
        C_t := (c_t1, ··· , c_tl) ← children of r_t;
        if k ≠ l then
            return false
        else
            r ← true;
            foreach c_mi in C_m and c_ti in C_t do
                r ← r∧ OccursAtRoot (c_mi, c_ti);
            return r
    else if (r_m, r_t) = ((*), ()) or ((), (*)) then
        return true
    else
        return r_m = r_t
```

We present in Figure 5.3 several examples to illustrate motif overlapping.

Definition 5.6 (Motif overlapping): Two motifs m_1 and m_2 *overlap* if

$$\forall \text{ subtree } t_1 \text{ of } m_1, \mathcal{R}_{t_1} \cap \mathcal{R}_{m_2} \neq \varnothing \vee \forall \text{ subtree } t_2 \text{ of } m_2, \mathcal{R}_{t_2} \cap \mathcal{R}_{m_1} \neq \varnothing.$$

We say a motif set is *overlap-free* if any two motifs in the set are not overlapped.

5.1.2 *Local defect*

Consider a motif m^* and a sequence w, we define the *local defect* $D^L(w, m^*)$ similarly as the structure defect D, by restricting the ensemble to the equivalence class of m^*. In other words, we replace $\mathcal{S}_n$ with the equivalence class $[m^*] := \{m \in \mathcal{M}_{|m^*|}; \psi(m) = \psi(m^*)\}$.

Definition 5.7 (Local defect): Given a motif m^* and a compatible sequence w, we define each local defect as,

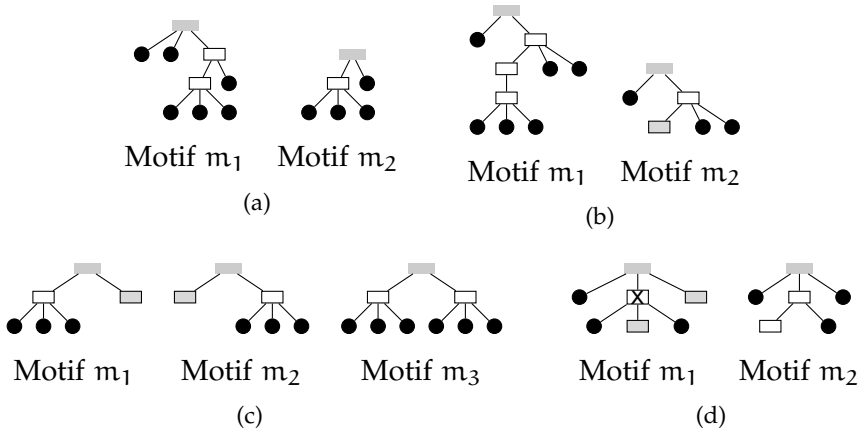


Figure 5.3: Examples where motifs m_1 and m_2 overlap.In both (a) and (b), motif m_1 strictly contains motif m_2.It is easy to see that, in (c), $\mathcal{R}_{m_1} \cap \mathcal{R}_{m_2} = \{m_3\}$.In (d), the subtree of m_1 at the x node occurs at the root of m_2.

1. *Local Suboptimal Defect*

$$D^{L,S}(w, m^*) := \min_{m \in [m^*] m \neq m^*} E(w, m^*) - E(w, m) \in \mathbb{R};$$

2. *Local Probability Defect*

$$D^{L,P}(w, m^*) := \sum_{\substack{m \in [m^*] \\ m \neq m^*}} \mathbb{P}^{[m^*]}(m \mid w) = 1 - \mathbb{P}^{[m^*]}(m^* \mid w) \in [0, 1];$$

3. *Local Ensemble Defect*

$$\begin{aligned} D^{L,E}(w, m^*) &:= \sum_{m \in [m^*]} \mathbb{P}^{[m^*]}(m \mid w) \cdot \mathrm{DPdist}(m, m^*) \\ &= |w| - \sum_{(i,j) \in m^*} 2p_w^{[m^*]}(i, j) - \sum_{i \text{ unpaired in } m^*} q_i^{[m^*]} \in \mathbb{R}_+. \end{aligned}$$

Note that motif probability $\mathbb{P}^{[m^*]}(m|w)$, base pair probability $p_w^{[m^*]}$, unpaired probability $q_i^{[m^*]}$ are also defined over the equivalence class of m^*. The following observation allows to extrapolate a family of undesignable structures from a (finite) collection of motifs.

Proposition 5.2 (Monotonicity): *For suboptimal or probability defect* $D \in \{D^S, D^P\}$, *sequence* w, $|w| = n$, *and structure* $S \in \mathcal{S}_n$, *one has*

$$\forall m \in S,\ D(w, S) \geqslant D^L(w_{[|m|]}, m)$$

where $w_{[|m|]}$ *is the restriction of* w *to the positions in* m.

Proof. Let $m \in S$ be a motif occurring in structure S. For each motif x in the equivalence class $[m]$ of m, we define the secondary structure S_x obtained by substituting x for m in S. One has

$$E(w_{[|m|]}, x) - E(w_{[|m|]}, m) = E(w, S_x) - E(w, S),$$

meaning that

$$\mathcal{B}(w_{[|m|]}, x)/\mathcal{B}(w_{[|m|]}, m) = \mathcal{B}(w, S_x)/\mathcal{B}(w, S).$$

Since the set $\mathcal{S}_m$ is finite, there exists a motif $\tilde{m}$ such that

$$\tilde{m} := \operatorname*{argmin}_{\substack{x \in \mathcal{S}_m \\ x \neq m}} E(w_{[|m|]}, x) - E(w_{[|m|]}, m).$$

We have,

$$D^{L,S}(w_{[|m|]}, m) = -(E(w_{[|m|]}, \tilde{m}) - E(w_{[|m|]}, m)) = -(E(w, S_{\tilde{m}}) - E(w, S)).$$

In addition, $\min_{x \in \mathcal{S}_n} E(w, x) - E(w, S) \leqslant E(w, S_{\tilde{m}}) - E(w, S)$, which implies the inequality $D^S(w, S) \geqslant D^{L,S}(w_{[|m|]}, m)$.

For the case $D = D^P$,

$$\begin{aligned}
\mathbb{P}(S \mid w) &= \frac{\mathcal{B}(w, S)}{\sum_{S' \in \mathcal{S}_n} \mathcal{B}(w, S')} \\
&\leqslant \frac{\mathcal{B}(w, S)}{\sum_{x \in [m]} \mathcal{B}(w, S_x)} \quad \text{since the set } \{S_x; x \in [m]\} \text{ is a subset of } \mathcal{S}_n \\
&= \frac{1}{\sum_{x \in [m]} (\mathcal{B}(w, S_x)/\mathcal{B}(w, S))} = \frac{1}{\sum_{x \in [m]} (\mathcal{B}(w_{[|m|]}, x)/\mathcal{B}(w_{[|m|]}, m))} \\
&= \frac{\mathcal{B}(w_{[|m|]}, m)}{\sum_{x \in [m]} \mathcal{B}(w_{[|m|]}, x)} = \mathbb{P}^{[m]}(m \mid w_{[|m|]}).
\end{aligned}$$

Therefore, $D^P(w, S) = 1 - \mathbb{P}(S \mid w) \geqslant 1 - \mathbb{P}^{[m]}(m \mid w_{[|m|]}) = D^{L,P}(w_{[|m|]}, m)$.

□

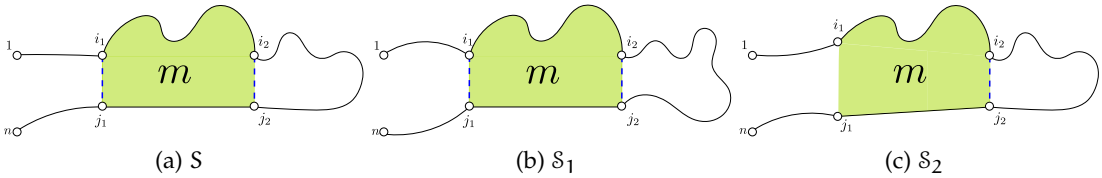
Intuitively, the same proposition should also be applicable for ensemble defects. However, it is not trivial to demonstrate it or to find a counterexample since the ensemble of interest changes while passing from motif to structure. We find out, for now, that a slightly modified proposition holds for the local ensemble defect.

Proposition 5.3 (Monotonicity of Ensemble Defect): *Given a sequence w, $|w| = n$, and structure $S \in \mathcal{S}_n$, one has*

$$\forall m \in S,\ D(w, S) \geqslant \min(D^L(w_{[|m|]}, m), 2)$$

where $w_{[|m|]}$ is the restriction of w to the positions in m.

While computing the structure ensemble defect, the motif's root and the open base pairs are no longer guaranteed to be paired (for example, positions (i_1, j_1) in Figure 5.4). We can separate structures into two sets depending on the presence of these base pairs. In the set where the appearance is preserved ($\mathcal{S}_1$ in the proof, Figure 5.4b), the region inside and outside the motif are independent. The structure ensemble defect defined in $\mathcal{S}_1$ is then the sum of ensemble defects defined in each region. The one for the inside region is exactly the local defect of motif. On the other hand, in the set $\mathcal{S}_n \setminus \mathcal{S}_1$ (Figure 5.4c), structures have a distance of at least 2 to the target one, but the relation to the local defect is unknown. Since the structure probability defined in $\mathcal{S}_n \setminus \mathcal{S}_1$ is not easy to determine, we cannot have a better lower bound than two despite the structures with larger distance.



(a) S (b) $\mathcal{S}_1$ (c) $\mathcal{S}_2$

Figure 5.4: Let S be a secondary structure containing the motif m (a) with the root base pair (i_1, j_1) and the open base pair (i_2, j_2). Alternative secondary structures having the same length as S are classified into two groups. In the first one (b), both (i_1, j_1) and (i_2, j_2) are paired. In the second one (c), at least one pair of bases are not paired, (i_1, j_1) for example. In the later case, structure has a distance of at least two to the structure S.

Proof. By definition, The structure esemble defect of S is

$$D^E(w, S) = \sum_{S' \in \mathcal{S}_n} \mathbb{P}(S' \mid w) \mathrm{DPdist}(S', S).$$

Let $m \in S$ be a motif occurring in structure S, (a_0, b_0) be its root, and $(a_1, b_1), \ldots, (a_{\delta_m}, b_{\delta_m})$ be its open paired leaves. In addition, let $\mathcal{S}_1$ be the set of secondary structures having base pairs $(a_0, b_0), \ldots, (a_{\delta_m}, b_{\delta_m})$ with the associated partition function $\mathcal{Z}_1 := \sum_{S' \in \mathcal{S}_1} \mathcal{B}(w, S')$. We can then rewrite the structure ensemble defect

$$\begin{aligned}
&D^E(w, S) \\
&= \sum_{S' \in \mathcal{S}_1} \mathbb{P}(S' \mid w) \mathrm{DPdist}(S', S) + \sum_{S' \in \mathcal{S}_n \setminus \mathcal{S}_1} \mathbb{P}(S' \mid w) \mathrm{DPdist}(S', S) \\
&= \frac{\mathcal{Z}_1}{\mathcal{Z}} \sum_{S' \in \mathcal{S}_1} \mathbb{P}_1(S' \mid w) \mathrm{DPdist}(S', S) + (1 - \frac{\mathcal{Z}_1}{\mathcal{Z}}) \sum_{S' \in \mathcal{S}_n \setminus \mathcal{S}_1} \mathbb{P}_2(S' \mid w) \mathrm{DPdist}(S', S)
\end{aligned}$$

where $\mathcal{Z} := \sum_{S' \in \mathcal{S}_n} \mathcal{B}(w, S')$ is the partition function over $\mathcal{S}_n$ and $\mathbb{P}_1(S' \mid w)$ (resp. $\mathbb{P}_2(S' \mid w)$) is the structure probability defined over $\mathcal{S}_1$ (resp. $\mathcal{S}_n \setminus \mathcal{S}_1$). Since a structure S' in the set $\mathcal{S}_n \setminus \mathcal{S}_1$ has at least two positions different to one of base pairs $\{(a_i, b_i)\}_{i \in \{0, \ldots, \delta_m\}}$, the distance to the target structure S is at least 2 in the second term. Therefore,

$$\sum_{S' \in \mathcal{S}_n \setminus \mathcal{S}_1} \mathbb{P}_2(S' \mid w) \mathrm{DPdist}(S', S) \geqslant 2 \sum_{S' \in \mathcal{S}_n \setminus \mathcal{S}_1} \mathbb{P}_2(S' \mid w) = 2.$$

Next, we will show that $\sum_{S' \in \mathcal{S}_1} \mathbb{P}_1(S' \mid w) \geqslant D^{L,E}(w_{[|m|]}, m)$. Let $\mathcal{S}_m$, a subset of $\mathcal{S}_1$, be the set of secondary structures containing motif m. Given a secondary structure S' in $\mathcal{S}_m$ and a motif x in the equivalent class of m, we use S'_x to denote the structure obtained by substituting m with x in S'. We have,

$$\mathcal{S}_1 = \bigcup_{S' \in \mathcal{S}_m} \{S'_x;\, x \in [m]\}.$$

Thus,

$$\begin{aligned}
\sum_{S' \in \mathcal{S}_1} \mathbb{P}_1(S' \mid w) \mathrm{DPdist}(S', S) &= \sum_{S' \in \mathcal{S}_m} \sum_{x \in [m]} \mathbb{P}_1(S'_x \mid w) \mathrm{DPdist}(S'_x, S) \\
&\geqslant \sum_{S' \in \mathcal{S}_m} \sum_{x \in [m]} \mathbb{P}_1(S'_x \mid w) \mathrm{DPdist}(x, m) \\
&= \sum_{S' \in \mathcal{S}_m} \frac{Z_{S'}}{Z_1} \sum_{x \in [m]} \mathbb{P}_{S'}(S'_x \mid w) \mathrm{DPdist}(x, m)
\end{aligned}$$

where $Z_{S'}$ and $\mathbb{P}_{S'}(S'_x \mid w)$ are the partition function and the structure probability defined over the set $\{S'_x;\, x \in [m]\}$ given a structure S' in $\mathcal{S}_m$. We have shown in the previous proof that $\mathbb{P}_{S'}(S'_x \mid w) = \mathbb{P}(x \mid w_{[|m|]})$, which gives

$$\begin{aligned}
\sum_{S' \in \mathcal{S}_1} \mathbb{P}_1(S' \mid w) \mathrm{DPdist}(S', S) &= \sum_{S' \in \mathcal{S}_m} \frac{Z_{S'}}{Z_1} \sum_{x \in [m]} \mathbb{P}^{[m]}(x \mid w_{[|m|]}) \mathrm{DPdist}(x, m) \\
&= \sum_{S' \in \mathcal{S}_m} \frac{Z_{S'}}{Z_1} D^{L,E}(w, m) \\
&= D^{L,E}(w_{[|m|]}, m).
\end{aligned}$$

Therefore,

$$D^{E}(w, S) \geqslant \frac{\mathcal{Z}_1}{\mathcal{Z}} \times D^{L,E}(w_{[|m|]}, m) + (1 - \frac{\mathcal{Z}_1}{\mathcal{Z}}) \times 2 \geqslant \min(D^{L,E}(w_{[|m|]}, m), 2).$$

□

Corollary 5.4: *If there exists a motif $m \in S^*$ such that*

$$\forall w \in \Sigma^{|m|},\ D^{L}(w, m) > \varepsilon,$$

then S^* cannot be D-designed. Note that ε is smaller than 2 for ensemble defect.

In other words the presence, in the target structure S^*, of a motif that cannot be designed *locally*, named *local obstruction*, is sufficient to forbid the existence of a sequence w that would constitute a design for S^*. Given a motif m, we use D_m to denote its *minimum possible defect*,

$$D_m := \min_{w \in \Sigma^{|m|}} D^{L}(w, m).$$

A local obstruction is then a motif with minimum possible defect surpasses the tolerance ε.

Proposition 5.5: *The minimum possible probability defect* D^P_m *and ensemble defect* D^E_m *of a motif* m *is* 0 *if and only if* m *is shallow motif.*

If motif m is not a shallow motif, there are at least two motifs in the ensemble for any valid assignment, motif m and the shallow motif $\psi(m)$. Thus, the probability defect is not null and the ensemble defect neither. On the other hand, assigning the same nucleotide to all unpaired bases in a shallow motif leads to an ensemble with exactly one motif, the shallow one. The value of the defect is then zero.

Motifs not respecting the minimum distance θ are local obstructions. All such motifs contain common motifs, called *trivial motifs*.

Definition 5.8 (Trivial motifs): Given a minimum distance θ, trivial motifs are motifs composing a root base pair and unpaired bases such that the amount of unpaired bases is less than θ.

For example, the trivial motifs introduced by the minimum distance $\theta = 3$ in the Turner energy model are `()`, `(·)`, and `(··)`. The local suboptimal defect and local probability defect of a trivial motif are, respectively, ∞ and 1. Since releasing the root base pair is always preferable, the local ensemble defect of a trivial motif is two bases. In this thesis, trivial motifs are not included in the motif set unless specifically mentioned.

5.2 LOCAL OBSTRUCTIONS

In this section, we present an algorithm to compute local obstructions for a predefined design objective.

Problem 3:
Input: Size k, defect D, and tolerance ε

Output: Local obstructions of size k such that the minimum possible probability defect D_m is larger than ε for each local obstruction m.

5.2.1 *Emulating a local defect with constraints*

The current RNA folding tool takes a valid secondary structure and a sequence as input, meaning that we cannot directly compute the local defect $D^L(w, m)$ for a given motif m and a sequence w. We address this issue with the help of *constrained folding*.

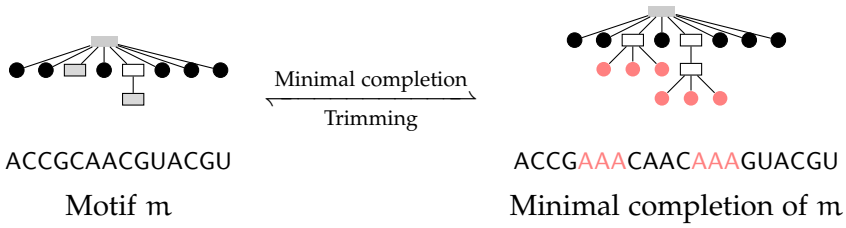


Figure 5.5: Minimal completion and trimming operation on a motif and its assignment.

Definition 5.9 (Folding constraint): Given a length k, a *folding constraint* C is a set consisting of positions from $[1, k]$ and pairs from $[1, k]^2$, respectively representing positions forced to remain unpaired and paired to a specific partner.

The conformation space is limited to structures compatible with the constraint C, denoted by $\mathcal{S}_C$ during the constrained folding. The *constrained defect* $D_C(w, S)$ can then be defined on the set $\mathcal{S}_C$, *i.e.*, replacing $\mathcal{S}_n$ with $\mathcal{S}_C$ in the original definition of defect. Such constraints are supported by reference implementations of energy minimization and partition function algorithms and can be easily enforced in simpler energy models.

Constrained folding allows us to extend a motif to a structure where the newly added part is constrained. In other words, only the bases involved in the motif are allowed to fold alternatively.

Definition 5.10 (Minimal completion): The *minimal completion* of a motif m for a nucleotide assignment w is a pair (S_m, w_m) such that:

- S_m is the secondary structure obtained from m by adding θ unpaired nodes (leaves) under each open paired leaf;
- w_m is the sequence obtained by inserting θ occurrences of the letter A under open paired leaves.

In the Turner model, the minimal completion is obtained by replacing each open paired leaf ▭ by ,

$S_m = m \circ (\underbrace{\cdots, \cdots}_{\delta_m})$.

Figure 5.5 illustrates the application of the minimal completion to a motif. A *trimming* operation is defined as the inverse of the completion and allows to recover a motif/sequence pair from its completion.

Definition 5.11 (Induced minimal constraint): Considering a motif m, and its minimal completion (S_m, w_m), we define the *induced constraint* C_m of m as consisting of:

- the root base pair of S_m;
- the base pairs in S_m stemming from the open paired leaves in m;
- the unpaired positions introduced by the completion.

Proposition 5.6: *For every defect* $D \in \{D^S, D^P, D^E\}$, *one has*

$$D^L(w, m) = D_{C_m}(w_m, S_m).$$

Proof. The schema of proof is similar to the one of Proposition 5.2. Let $\mathcal{S}_{C_m}$ be the set of secondary structures of length $|S_m|$ that are compatible with the folding constraint C_m. In fact, $\mathcal{S}_{C_m} = \{S_x; x \in [m]\}$, which is the set of minimal completion structures for motifs in the equivalent class of m. The energy of a structure $S_x \in \mathcal{S}_{C_m}$ is the sum of the energy contributed by the motif x and the one of the constrained part. Given a nucleotide assignment w for m, we have

$$E(w_m, S_x) - E(w_m, S_m) = E(w, x) - E(w, m)$$

for all $x \in [m]$ since the energy of constrained part is same for S_x and S_m. This proves the proposition for $D = D^S$. Moreover, one has

$$B(w_m, S_x)/B(w_m, S_m) = B(w, x)/B(w, m).$$

Thus, the structure probability defined in the set $\mathcal{S}_{C_m}$

$$\begin{aligned}
\mathbb{P}^{C_m}(S_m \mid w_m) &= \frac{B(w_m, S_m)}{\sum_{S_x \in \mathcal{S}_{C_m}} B(w_m, S_x)} \\
&= \frac{1}{\sum_{S_x \in \mathcal{S}_{C_m}} B(w_m, S_x)/B(w_m, S_m)} = \frac{1}{\sum_{x \in [m]} B(w, x)/B(w, m)} \\
&= \frac{B(w, m)}{\sum_{x \in [m]} B(w, x)} = \mathbb{P}^{[m]}(m \mid w)
\end{aligned}$$

and $D^{L,P}(w, m) = 1 - \mathbb{P}^{[m]}(w|m) = 1 - \mathbb{P}^{C_m}(S_m|w_m) = D^P_{C_m}(w_m, S_m)$.

Since the constrained part is same for any motif, we have $\mathrm{DPdist}(x, m) = \mathrm{DPdist}(S_x, S_m)$. Therefore, the equality for ensemble defect $D = D^E$.

□

It means that, in practice, the local defect of a motif can be computed by executing a constrained version of a global off-the-shelf algorithm (energy-minimization for D^S, base-pair probability for D^P and D^E) on the minimal completion of the motif.

5.2.2 *Computing local obstructions*

A naive solution for Problem 3 is to enumerate all motifs of size k and all compatible sequences for each, followed by a defect evaluation on their minimal completion. A motif is considered as local obstruction if all assignments fail the design condition. Let $P(k)$ be the time complexity of computing defect of a length k motif, which is usually polynomial. the time complexity of the naive approach is $\mathcal{O}(3^k 4^k P(k))$.

One can notice that motifs in the same equivalent class have compatible sequences in common, implying that each sequence is folded and evaluated several times in the previous approach. It is sufficient to fold each sequence once since only the MFE conformation in each equivalent class is considered in the design problem. Since each equivalence class is represented by one shallow motif, we can restrict the enumeration to shallow motifs. The complexity is reduced to $\mathcal{O}(\phi^{k-2} 4^k P(k))$ where $\phi = (1+\sqrt{5})/2 \approx 1.618$ is the golden ratio, the exponential constant for Fibonacci numbers. Indeed, one can observe that shallow motifs are in bijection with 1D tilings using monominoes (unpaired bases) and dominoes (open base pairs), so that the number of shallow motifs of size $k+2$ coincides with the k-th Fibonacci number.

Given a defect D restricted to a value ε, and a motif of length k, our algorithm executes the following steps:

- Enumerate all *shallow motifs* (of depth 1) of length k;
- For any such motif m°, consider any assignment w° consistent with the paired nodes in m°:
  - Build the minimal completion (S°_m, w°_m) of (m°, w°), and execute on w°_m a constrained MFE folding algorithm, using the induced constraint C_{m°;
  - If the MFE computation returns a unique structure $m^\star$, consider the motif m obtained by trimming $m^\star$ ($m \in [m^\circ]$);
  - Evaluate the local defect and, if $D^L(w^\circ, m) \leqslant \varepsilon$, add m to the list $\mathcal{D}_k$ of designable motifs;
- Return $\mathcal{M}_k$, the set of all motifs of length k not in $\mathcal{D}_k$.

A detailed version of the procedure is described in Algorithm 5.2 and illustrated in Figure 5.6.

Proposition 5.7: *All motifs returned by Algorithm 5.2 are local obstructions.*

Proof. First, let us consider the properties of a motif m returned by the algorithm. Since $m \notin \mathcal{M}$ then, for each sequence w°, either a lower constrained MFE fold was found, or the local defect exceeded ε. In the latter case, Propositions 5.2 and 5.3 imply that any pair (S, w), where S features m, and sequence w having nucleotide assignment w° on the motif positions, has defect greater than ε, thus w is not a

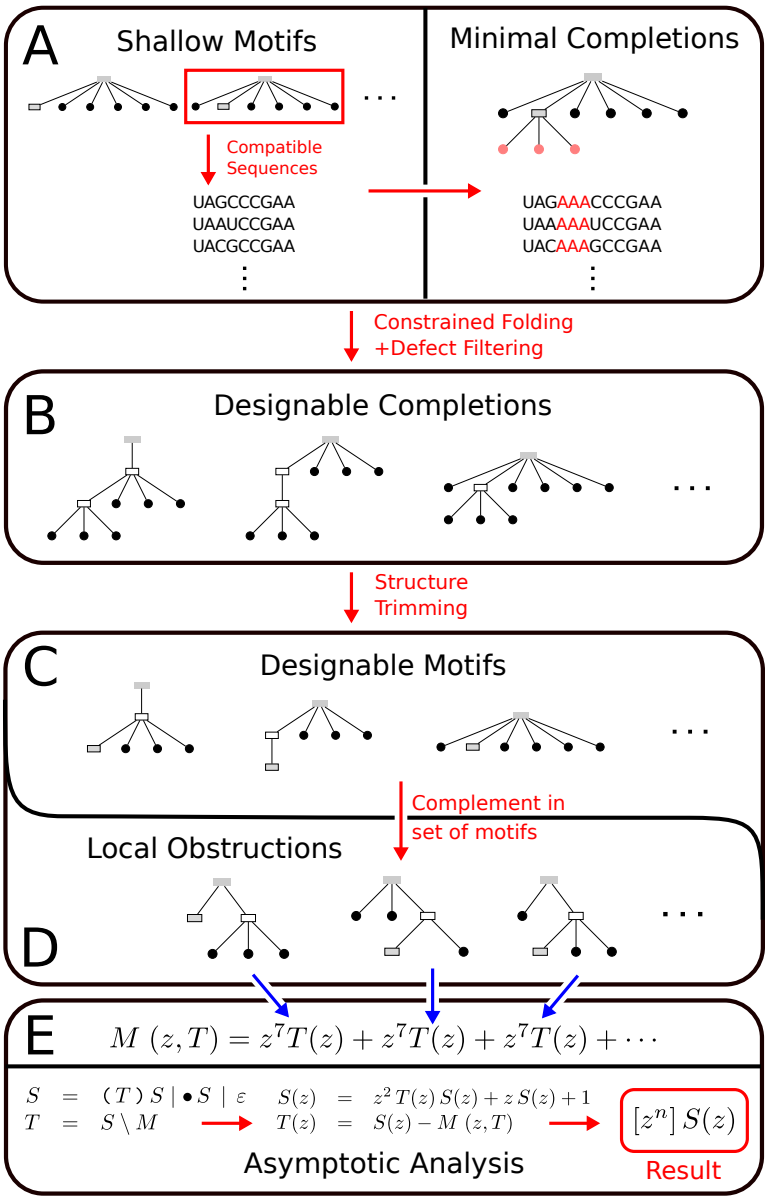


Figure 5.6: Workflow for undesignable motif identification.

design for S. In the former case where an alternative motif m^a is preferred to (or equally stable as) m for w°, then for any structure S containing m and sequence w, having nucleotide assignment w° on the positions of m, a competitor to S for w can be constructed by replacing m by m^a in S. One concludes that, if $m \notin \mathcal{M}$, any structure S, $m \in S$, and sequence w does not represent a (D, ε)-design. □

5.2.3 *Extracting an overlap-free subset of motifs*

The analyses in the following chapters require motif set to be overlap-free, meaning that some motifs obtained from Algorithm 5.2 should be discarded. Moreover, we

Algorithm 5.2: Computing local obstructions of a given size

```
Input  : A motif size k, an energy model E, a defect definition D, and a tolerance ε
Output: M_k a, possibly empty, set of local obstructions of length k
D_k ← ∅;
Q_k ← Set of all shallow motifs of size k;
foreach m° ∈ Q_k do
    foreach w° ∈ Σ^k (compatible with m°) do
        C_m° ← Induced constraint of m°;
        (S°_m, w°_m) ← Minimal completion of (m°, w°);
        O ← MFE(w_m | C_m°) w.r.t. energy model E;
        if O = {S'} then
            if D_{C_m°}(w°, S') ⩽ ε then
                m ← trimming of S' ;
                D_k ← D_k ∪ {m}
A_k ← Complete set of all motifs of size k;
return M_k = A_k − D_k
```

want to keep as many local obstructions as possible in the original set to have a closer asymptotic estimation. Formally, given a motif set $\mathcal{M}$ and a list of overlapped motif pairs $\mathcal{P} = \{(m_1, m_2)\} \in \mathcal{M}^2$, *i.e.*, motifs m_1 and m_2 overlap as defined in Definition 5.6, we aim to extract a subset $\mathcal{M}^* \subseteq \mathcal{M}$ such that any two motifs in $\mathcal{M}^*$ are not in $\mathcal{P}$ while maximizing $|\mathcal{M}^*|$. It is equivalent to obtain a Maximum Independent Set (MIS) of the graph $G = (\mathcal{M}, \mathcal{P})$. However, finding MIS of a general graph is an NP-hard problem [33], which implies the absence of an efficient solution for our problem.

To work around the hardness, we used an $\mathcal{O}(|\mathcal{M}|/(\log(\mathcal{M}))^2)$-approximation algorithm to extract an overlap-free motif set [9]. In addition, we prefer a smaller motif or a higher defect per length motif, depending on the demand. Therefore, our heuristic approach applies a pre-processing on the motif set before calling the MIS algorithm, which results in three steps,

1. Any motif that strictly contains another motif is removed;
2. For each pair of motifs in increasing lexicographic order, if two motifs are overlap, then
    - the larger one is removed in structure enumerating problem in Chapter 6;
    - the one with lower defect per length is eliminated in structure defect estimating in Chapter 7;
3. We build the graph $G = (\mathcal{M}, \mathcal{P})$ described above over the remaining motifs and extract the MIS using Boppana and Halldórsson algorithm [9].

Defect	ε	$\lvert\mathcal{M}_{D,\varepsilon}\rvert$	$\lvert\mathcal{F}_{D,\varepsilon}\rvert$
D^S	0	4 561	387
D^S	-1	7 000	573
D^P	0.5	4 845	401
D^P	0.1	7 305	611
D^P	0.01	8 187	709
D^E	1	5 012	411

Table 5.1: Collections of local obstructions of length up to 14.

5.2.4 *Local obstruction database*

We implemented Algorithm 5.2 in Python3 and executed with minimum distance $\theta = 3$ for motif length up to 14 bases and different design objectives. There are 10 886 motifs of length up to 14, in which 3 070 of them do not contain isolated base pair, and 606 are shallow motifs. We computed local obstructions, denoted by $\mathcal{M}_{D\leqslant\varepsilon}$, for a given defect D and a threshold ε. Then, we filtered $\mathcal{M}_{D\leqslant\varepsilon}$ as described in Section 5.2.3 to extract an overlap-free set, denoted by $\mathcal{F}_{D\leqslant\varepsilon}$. Table 5.1 summarizes the size of each local obstruction collection.

OBSTRUCTIONS TO BASIC INVERSE FOLDING ($D^S \leqslant 0$) In the classic RNA design setting, the inverse folding, one attempts to design a sequence that admits a target structure as its unique MFE structure. The setting corresponds to choosing the suboptimal defect D^S with $\varepsilon = 0$.

Our analysis of motifs for a length up to 14 reveals 4 561 local obstructions out of 10 886 motifs, meaning that almost half of motifs are not designable. The amount of local obstructions grows exponentially with length, as can be seen in Figure 5.7. Among these local obstructions, 60 of them are undesignable because of the existence of another MFE conformation. In addition, an overwhelming majority (4 490 out of 4 561) of which contain at least one isolated base pair. Figure 5.8 shows some local obstructions identified without isolated base pair. The nearest-neighbor energy model expects such motifs to be heavily penalized, yet not explicitly forbidden (unless specified). An overlap-free set of 387 local obstructions was obtained after motif filtering.

Consecutive bulges, alternating on the 5′ and 3′ ends of a helix, also seem systematically suboptimal for the Turner energy model. A large interior loop being systematically favored as a candidate for the MFE. Finally, hairpin/terminal loops directly stemming from multi-loops are systematically discriminated, and a structure consisting of a larger unpaired stretch in the multi-loop will always be favored.

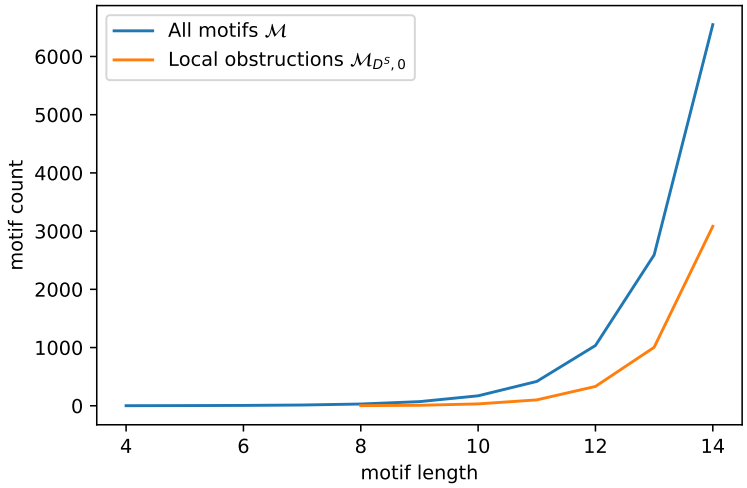


Figure 5.7: Amount of motifs (blue) and local obstructions (orange) for different length.

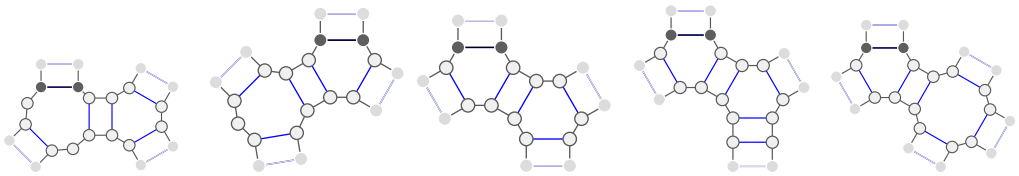
OBSTRUCTIONS WITH SUPEROPTIMAL DESIGN OBJECTIVE ($D^S \leqslant -1$) We also consider a more challenging design objective. The target structure must be the MFE of a sequence and be superoptimal, *i.e.*, achieve an energy distance to its first suboptimal at least 1 kcal.mol^{-1}. This objective corresponds to using the suboptimal defect D^S with $\varepsilon = -1$.

Under these more substantial constraints, with 2 439 newly determined obstructions, the number of local obstructions increases to 7 000 motifs and 573 for the overlap-free set. The majority still contains at least one isolated base pair. Only 144 out of 7 000 obstructions do not feature isolated base pairs. Furthermore, most novel obstructions have a suboptimal defect around -0.4 and -0.7 kcal.mol^{-1}, as seen in Figure 5.9.

OBSTRUCTIONS WITH SMALL EQUILIBRIUM PROBABILITIES Being MFE conformation may have a low Boltzmann probability. Next, we turn to the probability defect D^P, and investigate the impact of ε. We consider 3 values for the threshold $\varepsilon \in \{0.5, 0.1, 0.01\}$, associated with targeted Boltzmann probabilities for the motifs greater than 50%, 90% and 99% respectively, and show in Figure 5.10 some local obstructions under these conditions. Figure 5.11 presents the defect distribution of obstructions with local probability defect beyond 0.01. Most of the motif minimum possible defects are around 0.01 and from 0.2 to 0.4.

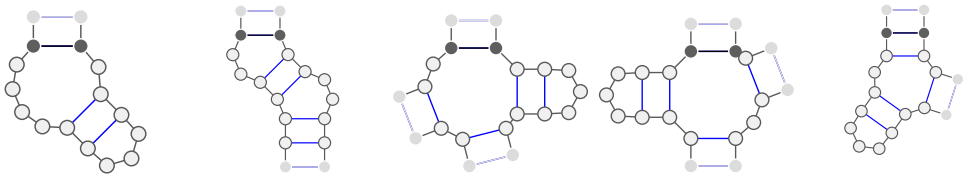
For threshold $\varepsilon = 0.5$, the size of the obstruction set slightly increases from 4 561, the one for the classic design, to 4 845 motifs, and we have $\mathcal{M}_{D^S \leqslant 0} \subset \mathcal{M}_{D^P \leqslant 0.5}$. It is not entirely unexpected since our definition of a valid design requires the target motif to be the sole MFE for the sequence. We obtain a collection of 7 305 obstructions for $\varepsilon = 0.1$ and 8 187 obstructions for $\varepsilon = 0.01$. Under extreme strict design objective ($\varepsilon = 0.01$), about three forth of motifs are considered undesignable, and only 464 obstructions do not feature isolated base pairs, which confirms that the stacking

Distance to subopts $\Delta \geq$ 0 kcal.mol^{-1} ($D^S \leq 0$).



ID	m17e8c	md73a	m17eb	m177ab	m5feb

Distance to subopts $\Delta \geq$ 1 kcal.mol^{-1} ($D^S \leq -1$).



ID	m140a00	m141ea0	m502bc	mcd40a	m1d40a2
Defect	-0.4	-0.5	-0.5	-0.5	-0.8

Figure 5.8: Some local obstructions without isolated base pair for suboptimal defect D^S with tolerance $\varepsilon \in \{0, -1\}$. Each local obstruction is associated with an ID based on 64-based encoding and a defect value if any.

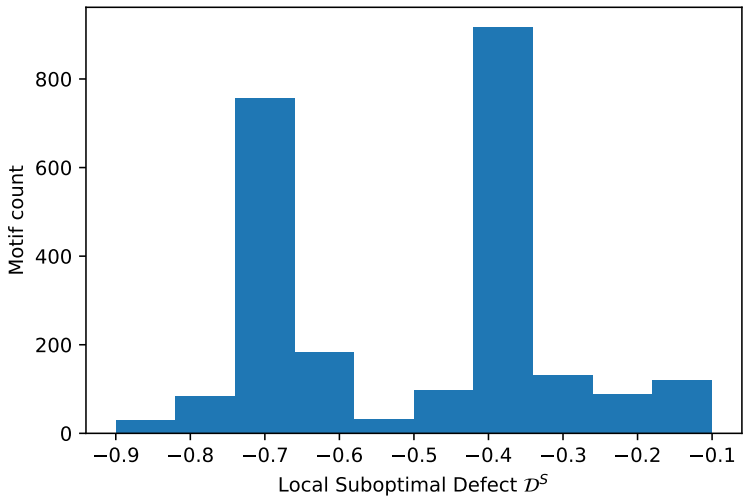
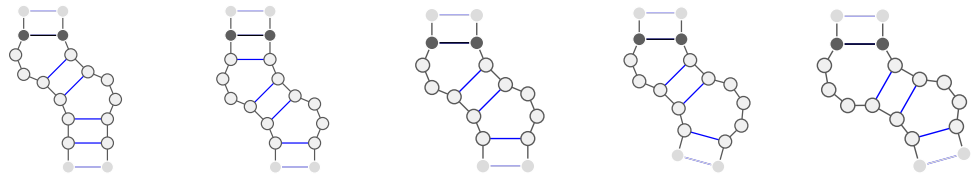


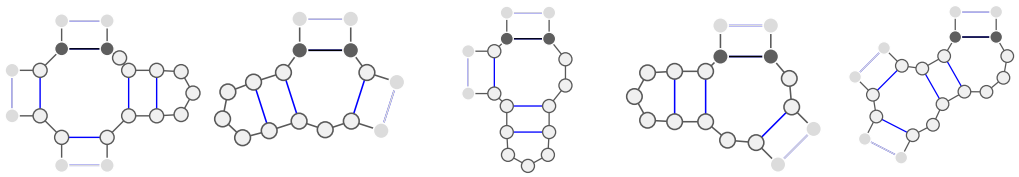
Figure 5.9: Histogram of suboptimal defect of newly determined obstructions within design objective ($D^S \leqslant -1$).

Undesignable motifs with probability defect $D^P \leq$ 50%.



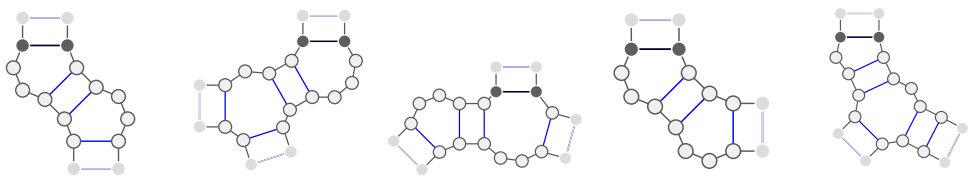
ID	m141ea0	m150e82	m143a0	m503a0	m50e80
Defect	0.63634	0.63634	0.61914	0.50879	0.50879

Probability defect $D^P \leq$ 10%.



ID	m140af	m3140a	m502b	m3140a	m14fa
Defect	0.35039	0.269	0.2429	0.13785	0.1287

Probability defect $D^P \leq$ 1%.



ID	m143a0	m17ca	mc170a	m170a0	m51ee8
Defect	0.61914	0.13781	0.08649	0.03255	0.01023

Figure 5.10: Some local obstructions without isolated base pair for probability defect D^P with tolerance $\varepsilon \in \{0.5, 0.1, 0.01\}$. Each local obstruction is associated with an ID based on 64-based encoding and its local probability defect value.

loop stabilizes the secondary structure. Upon filtering, we identify 401, 611, and 709 overlap-free local obstructions for $\varepsilon = 0.5$, 0.1, and 0.01 respectively.

OBSTRUCTIONS HAVING LARGE EXPECTED EQUILIBRIUM DISTANCE Last but not least, we consider a design objective where the target structure should be central at the thermodynamic equilibrium. To that purpose, we enforce that, within the Boltzmann distribution, the expected base-pair distance to the target structure remains smaller than one base pair. This corresponds to using the ensemble defect D^E with threshold $\varepsilon = 1$.

Following the same procedure as above, some undesignable motifs are identified, with ensemble defects exceeding 1. It leads to a larger set of local obstructions (5 012 in total) than the classic inverse folding, with 122 motifs devoid of isolated base pairs

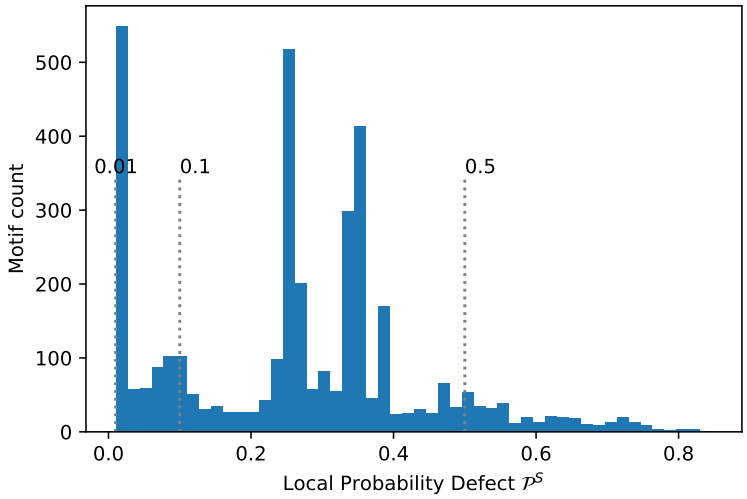
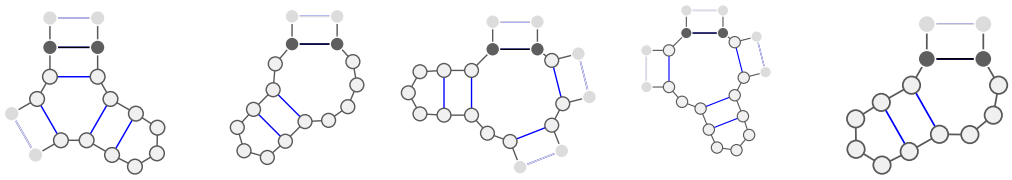


Figure 5.11: Distribution of motif minimum possible probability defects truncated at 0.01.

Undesignable motifs with expected distance ≤ 1 bp ($D^E \leq 1$).



ID	mLVwA	m7+N-k	mGr+g	mHVwo	mVr+g
Defect	1.51621	1.38023	1.3797	1.26794	1.02468

Figure 5.12: Some local obstructions without isolated base pair for ensemble defect D^E with tolerance $\varepsilon = 1$. Each local obstruction is associated with an ID based on 64-based encoding and its local ensemble defect value.

that some of them are shown in Figure 5.12. The extracted overlap-free set contains 411 obstructions. Obstructions previously determined for the design objective ($D^P \leqslant 0.5$) are also undesignable for this one, *i.e.*, $\mathcal{M}_{D^P \leqslant 0.5} \subset \mathcal{M}_{D^E \leqslant 1}$. Designs for such motifs have more than 50% chance of folding into alternative conformations other than the target one. It gives an expected distance of more than one in the ensemble since alternative conformation has at least two bases differently paired.

5.3 UNDESIGNABLE AND HARD MOTIFS IN EXPERIMENTALLY-DETERMINED 3D STRUCTURES

In this section, we present an analysis of experimentally-determined RNA 3D structures, and investigate whether motifs considered as undesignable, or hard (defect-inducing), with respect to the Turner energy model, can be found in existing structures.

DATASETS DESCRIPTION. The Protein Data Bank (PDB) [11] is a global repository that collects 3D models of biological molecules derived from experiments. At the date of our study (March 2021), it contained a total of 5 313 PDB entries containing at least one RNA chain, which were downloaded. We then used the DSSR v2.2.1 software [54] to annotate the secondary structures, pseudoknots, intermolecular base pairs, and non-canonical base pairs. We removed entries lacking base pairs, obtaining a dataset of 4 543 PDB entries, including a total of 1 634 067 canonical base pairs.

METHOD. For any given motif m and secondary structure S, we used a simple pattern matching algorithm in $\Theta(|S| \times |m|)$ to find all occurrences of m. Positions involving tertiary base pairs (non-canonical base pairs + pseudoknots + intermolecular base pairs) were treated as unpaired during the search, but their presence was memorized. In addition, pattern matching is run independently on each RNA strand in the same PDB entry.

5.3.1 *Occurrences of undesignable motifs*

Firstly, we investigate possible occurrences, within 3D structures, of undesignable motifs for the classic inverse folding, *i.e.*, suboptimal defect $D^S \leqslant 0$. We consider a collection of 4 561 undesignable motifs $\mathcal{F}_{D^S \leqslant 0}$ obtained in Section 5.2.4 of length from 8 up to 14. The occurrence of such motifs would be surprising since, in the Turner energy model, such motifs are expected to adopt alternative, more stable, conformations at the thermodynamic equilibrium.

A systematic search for undesignable motifs in the PDB revealed a total of 16 986 occurrences of 221 distinct motifs. Those occurrences are concentrated in 1 332 PDB entries out of the possible 4 543, meaning that 80% of the PDB entries avoid all undesignable motifs. Given the total number of 10 886 motifs of size up to 14 and (very roughly) assuming a uniform probability of occurrences for all motifs, one would expect around 684 000 occurrences of undesignable motifs in our PDB dataset in the absence of any negative bias. The significant difference to the expected occurrence amount suggests a bias against occurrences of undesignable motifs, consistent with their relative instability and negative selective pressure.

Moreover, among all 16 986 motif occurrences, 16 650 feature additional non-canonical Base Pairs (ncBPs) involving at least one nucleotide in the occurrence, and pseudoknots present in 43 of them. The overwhelming proportion of ncBPs within occurrences of undesignable motifs supports the hypothesis that ncBPs stabilize undesignable motifs, leading to their adoption in timescales that are compatible with their observation in 3D structures.

Finally, we focused on the remaining occurrences of forbidden motifs, devoid of ncBPs, pseudoknots or interactions. We were left with 336 occurrences, over 80 distinct motifs, which we further analyzed. We discarded motifs featuring less than 5 occurrences, and also those strictly extending another motif in the list. Five motifs remained warranting further analysis.

Motif `((...).)` is the most observed motif with 38 occurrences, mainly found in HIV-1 Trans-Activation Response (TAR), Iron response element, and the ribosome. Instead of forming a hairpin loop of 6 unpaired bases, the base pair at position $(2, 5)$ separates the loop into a bulge and a hairpin loop of 3 nucleotides. In 36 out of 38 occurrences, the base pair $(2, 5)$ is CG base pair, which has been shown experimentally to stabilize the structure of HIV-1 TAR and help the promoter activation in the presence of protein Tat [49].

The second most occurred motif `((...)..)` with 11 occurrences has one unpaired position more. Half of the occurrences were found in Cas12i2, a class two CRISPR, complex. The bulge region has been recognized to interact with the WED domain of Cas12i2 [44]. The remaining three motifs are largely found in the (mito)ribosome. Motif `..((*).))` (10 occurrences) is mainly found in mitochondrially encoded 12S ribosomal RNA (12s rRNA), 4 out 7 `(.(...))` occurrences are in 23s rRNA, and `(.((*)..))` (6 occurrences) appears mostly in 18s rRNA.

5.3.2 *Motifs defect correlates negatively with their number of occurrences*

Next, we want to study the correlation between motif occurrences and motif design hardness, *i.e.*, defect. We consider the complete set of 10886 motifs of length up to 14, each associated with its lowest possible ensemble defect. The set includes 606 shallow motifs and 4 501 forbidden motifs, a motif that is not the MFE folding for any compatible sequences ($D^S \leqslant 0$). The ensemble defect of a forbidden motif is considered non-determined.

We found 2 083 different motifs in experimentally determined structures with at least one occurrence, including 238 shallow motifs and 212 forbidden motifs. The Spearman correlation between motif ensemble defect and occurrences omitting shallow and forbidden motifs is -0.29, which indicates a negative correlation between the two variables. Figure 5.13 presents the motif occurrence distribution in boxenplot of different ranges of ensemble defects. Since most motifs have a small value of ensemble defect, we used a varied and increasing bin size for the reason of visualization. It is clear to see a decreasing tendency of occurrence number while increasing the value of ensemble defect. In addition, setting the value of the forbidden motif at 1 nt also matches such tendency, which validates the choice for further analysis in Chapter 7.

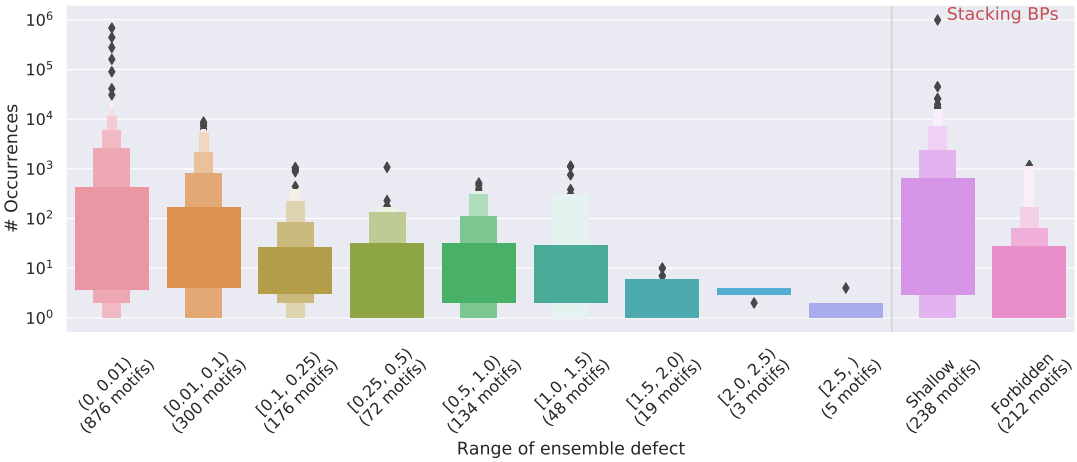


Figure 5.13: Boxenplot of motif occurrences for different range of ensemble defect. Starting from the median, each level in boxenplot (letter-value plot) contains half of the remaining data points [43].

6

ENUMERATING SECONDARY STRUCTURES AVOIDING LOCAL OBSTRUCTIONS

After identifying local obstructions, we wonder about the impact on designable secondary structures. In this section, we are interested in describing and enumerating designable secondary structures $\mathcal{D}$ for a defect D and a tolerance ε. Let $\mathcal{F}$ be an overlap-free set of local obstructions obtained using the previously described algorithm and $\mathcal{S}_{\bar{\mathcal{F}}}$ be the set of secondary structures avoiding $\mathcal{F}$,

$$\mathcal{S}_{\bar{\mathcal{F}}} = \{S \in \mathcal{S}; \forall m \in \mathcal{F},\ m \notin S\}.$$

Propositions 5.2 and 5.3 imply that its cardinality $|\mathcal{S}_{\bar{\mathcal{F}}}|$ sets up an upper bound for the number of designable secondary structures, $|\mathcal{D}| \leqslant |\mathcal{S}_{\bar{\mathcal{F}}}|$. Therefore, this section aims to answer the following problem.

Problem 4 (Designable Structures Counting):
Input: An overlap-free local obstruction set F for design objective $D \leqslant \varepsilon$

Output: Asymptotic value of s_n, the number of secondary structures of length n avoiding $\mathcal{F}$ with

$$s_n := |\{S \in \mathcal{S}_{\bar{\mathcal{F}}};\ |S| = n\}|\,.$$

Let $\mathcal{P}$ be a finite set of possibly-overlapping unlabelled trees, named patterns. Chyzak *et al.* showed that the joint distribution of pattern occurrences in a rooted labeled tree is a multivariate Gaussian limiting distribution [13]. The number of trees avoiding $\mathcal{P}$ can be computed by setting the number of occurrences at 0 for each pattern. The approach is based on establishing relations among all possible subtree patterns from $\mathcal{P}$ and marking the root occurrence of pattern in $\mathcal{P}$.

An algorithm is also proposed to mark and count pattern occurrences in an ordered tree by memorizing partial patterns seen [14]. The algorithm can be further transformed into a system of functional equations for asymptotic analysis. Unfortunately, in both cases, the system becomes complicated when the pattern set is huge. For instance, an overlapped obstruction set contains more than 4 000 motifs. In addition, the notion of tree size is different in our case. The length of a secondary structure differs to the number of nodes in its tree presentation since the length of a paired node is 2. For these reasons, we limit our study to overlap-free obstruction sets.

6.1 GRAMMAR AND GENERATING FUNCTION

We adopt the classic symbolic method as our approach [30]. First, we describe a grammar that generates objects in the set $\mathcal{S}_{\bar{\mathcal{F}}}$. Next, we turn the grammar into a system of functional equations, including the Ordinary Generating Function (OGF) of the set,

$$S_{\bar{\mathcal{F}}}(z) = \sum_{S \in \mathcal{S}_{\bar{\mathcal{F}}}} z^{|S|} = \sum_{n \geqslant 0} s_n z^n.$$

We start with the classic rule that generates all secondary structures with respect to a minimal distance $\theta = 0$. We introduce an additional rule while encountering a base pair since the root base pair always encloses a local obstruction. The second rule builds the set of structures $\mathcal{T}_{\bar{\mathcal{F}}} \subset \mathcal{S}_{\bar{\mathcal{F}}}$ that are still in $\mathcal{S}_{\bar{\mathcal{F}}}$ while completing with an external base pair,

$$\mathcal{T}_{\bar{\mathcal{F}}} = \{S \in \mathcal{S}_{\bar{\mathcal{F}}}; (S) \in \mathcal{S}_{\bar{\mathcal{F}}}\}.$$

The difference between two sets, $\mathcal{S}_{\bar{\mathcal{F}}}$ and $\mathcal{T}_{\bar{\mathcal{F}}}$, is the set of structures containing a local obstruction at the root in the presence of the external base pair,

$$\mathcal{S}_{\bar{\mathcal{F}}} \setminus \mathcal{T}_{\bar{\mathcal{F}}} = \bigcup_{m \in \mathcal{M}} \{m' \circ (T_1, \cdots, T_{\delta_m}); (T_1, \cdots, T_{\delta_m}) \in \mathcal{T}_{\bar{\mathcal{F}}}^{\delta_m}\}$$

with m' denotes the part delimited by the root of a motif m. In other words, the second rule subtracts structures having a root occurrence of m'.

Proposition 6.1 (Grammar describing $\mathcal{S}_{\bar{\mathcal{F}}}$): *Grammar below generates the set of secondary structures $\mathcal{S}_{\bar{\mathcal{F}}}$ avoiding an overlap-free set $\mathcal{F}$ of local obstructions,*

$$S_{\bar{\mathcal{F}}} \to (\ T_{\bar{\mathcal{F}}}\ )\ S_{\bar{\mathcal{F}}} + \bullet\, S_{\bar{\mathcal{F}}} + \varepsilon$$
$$T_{\bar{\mathcal{F}}} \to S_{\bar{\mathcal{F}}} - \sum_{m \in \mathcal{T}} R_{m'}$$

with $R_{m'} = m' \circ (T_{\bar{\mathcal{F}}}, \ldots, T_{\bar{\mathcal{F}}})$ where each open paired leaf of m' is extended with a non-terminal symbol $T_{\bar{\mathcal{F}}}$.

Proof. For the sake of readability, we abuse the notation $\mathcal{S}_n$ in this proof. Denote by $\mathcal{S}_n$ the real set of all secondary structures (not the generated ones) having length n and *avoiding* local obstructions in $\mathcal{F}$, and by $\mathcal{T}_n$ the ones of length n which are surrounded by an extra pair of parentheses and also avoid local obstructions in $\mathcal{F}$. Denote by $\mathcal{L}_S$ (resp. $\mathcal{L}_T$) the language generated from $S_{\bar{\mathcal{F}}}$ (resp. $T_{\bar{\mathcal{F}}}$), and by $\mathcal{L}_{Sn}$ (resp. $\mathcal{L}_{Tn}$) its restriction to secondary structures of length n.

We are going to prove the proposition by showing that for all $n \geqslant 0$, we have

$$\mathcal{L}_{Sn} = \mathcal{S}_n \text{ and } \mathcal{L}_{Tn} = \mathcal{T}_n. \tag{6.1}$$

This property immediately implies that

$$|\mathcal{L}_{Sn}| = s_n := |\mathcal{S}_n| \qquad \text{and} \qquad |\mathcal{L}_{Tn}| = t_n := |\mathcal{T}_n|.$$

First of all, Equation 6.1 holds for $n = 0$. It is easy to see that the empty structure ε avoids local obstructions and is the only element in $\mathcal{S}_0$, $\mathcal{S}_0 = \{\varepsilon\}$. In addition, the only word of length 0 generated from $S_{\bar{\mathcal{F}}}$ is ε, via the rule $S_{\bar{\mathcal{F}}} \to \varepsilon$. Thus, $\mathcal{L}_{S0} = \{\varepsilon\} = \mathcal{S}_0$. The case of $T_{\bar{\mathcal{F}}}$ is very similar, but depends of the presence/absence, in $\mathcal{F}$, of the local obstruction $m_\varepsilon = ()$ that pairs two consecutive bases. One has $\mathcal{T}_0 = \varnothing$ if $m_\varepsilon \in \mathcal{F}$, or $\{\varepsilon\}$ otherwise. As for the grammar, $T_{\bar{\mathcal{F}}} \to S_{\bar{\mathcal{F}}} \rightsquigarrow \varepsilon$ produces a word of length 0, but is subtracted by $T_{\bar{\mathcal{F}}} \to R_\varepsilon \rightsquigarrow \varepsilon$ if $m_\varepsilon \in \mathcal{F}$. Thus,

$$\mathcal{L}_{T0} = \left\{ \begin{array}{ll} \{\varepsilon\} & \text{if } m_\varepsilon \notin \mathcal{F} \\ \varnothing & \text{otherwise} \end{array} \right\} = \mathcal{T}_0.$$

Next, we assume that Equation 6.1 holds for any $n < k$, where k is a positive integer. Let S^* be a secondary structure of size k, *i.e.* $S^* \in \mathcal{S}_k$. Its first base is either paired with a base at position l or unpaired. We are going to show that, in both cases, S^* is a word of $\mathcal{L}_{S_k}$.

- In the paired case, S^* is the form $(T')S'$ with $T' \in \mathcal{T}'_{l-2}$ and $S' \in \mathcal{S}_{k-l}$. By the induction condition, T' and S' can both be generated from the grammar. It means that S^* can be produced as follow, $S_{\bar{\mathcal{F}}} \to (T_{\bar{\mathcal{F}}})S_{\bar{\mathcal{F}}} \rightsquigarrow (T')S' = S^*$ with $T' \in \mathcal{L}_{Tl-2}$ and $S' \in \mathcal{L}_{Sk-l}$. In addition, S^* is a word in $\mathcal{L}_{Sk}$ since its length $|S^*| = 2 + |T'| + |S'| = k$.

- On the other hand, S^* is the form $\bullet S''$. It is easy to see that $S'' \in \mathcal{S}_{k-1}$ and $S'' \in \mathcal{L}_{Sk-1}$ by the induction condition. Therefore, S^* is a word of $\mathcal{L}_{Sk}$ that is obtained via $S_{\bar{\mathcal{F}}} \to \bullet S_{\bar{\mathcal{F}}} \rightsquigarrow \bullet S'' = S^*$.

This proves the completeness of the first rule $S_{\bar{\mathcal{F}}} \to (\, T_{\bar{\mathcal{F}}} \,)\, S_{\bar{\mathcal{F}}} + \bullet\, S_{\bar{\mathcal{F}}} + \varepsilon$.

Let $S^* \in \mathcal{L}_{Sk}$ be a word generated from $S_{\bar{\mathcal{F}}}$. We will prove the correctness by showing that S^* is a valid secondary structure in $\mathcal{S}_k$, *i.e.* a structure of length k avoiding local obstructions. The word S^* is produced from the non-terminal symbol $S_{\bar{\mathcal{F}}}$ via one of the follows,

- $S_{\bar{\mathcal{F}}} \to (T_{\bar{\mathcal{F}}})S_{\bar{\mathcal{F}}} \rightsquigarrow (T')S' = S^*$, where $T' \in \mathcal{L}_{Tn_{T'}}$ and $S' \in \mathcal{L}_{Sn_{S'}}$ with $n_{T'} + n_{S'} = k - 2$. By induction condition, T' and S' are valid secondary structures in, respectively, $\mathcal{T}_{n_{T'}}$ and $\mathcal{S}_{n_{S'}}$, which means that both (T') and S' avoid local obstructions. $S^* = (T')S'$ avoids also local obstructions and is then a valid secondary structure in $\mathcal{S}_k$.

- $S_{\bar{\mathcal{F}}} \to \bullet S_{\bar{\mathcal{F}}} \rightsquigarrow \bullet S'' = S^*$ with $S'' \in \mathcal{L}_{Sk-1}$. S'' is a valid secondary structures in $\mathcal{S}_{k-1}$ by induction condition that do not contain local obstructions. Therefore, S^* neither.

We have shown that the completeness and the correctness of the first rule, which implies that $\mathcal{S}_n = \mathcal{L}_{Sn}$.

Let T^* be a secondary structure in $\mathcal{T}_k$ such that structure (T^*) avoids local obstructions, meaning that T^* does not contain an inner local obstruction at root. Thus, T^* is not a structure in any $\mathcal{R}_{m'}$ and is then a word of $\mathcal{L}_{Tk}$ generated via $T_{\bar{\mathcal{F}}} \to S_{\bar{\mathcal{F}}} \rightsquigarrow T^*$.

Conversely, let $T^* \in \mathcal{L}_{Tk}$ be a word produced from the non-terminal symbol $T_{\bar{\mathcal{F}}}$. We are going to proof that structure (T^*) does not contain a local obstruction. The word T^* is first generated using the rule $T \to S$. In other words, T^* does not contain local obstruction. Next, there are two cases,

- None of inner local obstruction occurs at the root of T^*. The secondary structure (T^*) also avoids local obstruction. Thus, $T^* \in \mathcal{T}_k$.
- Otherwise, the rule $T_{\bar{\mathcal{F}}} \to R_{m'} = m' \circ (T_{\bar{\mathcal{F}}}, \dots, T_{\bar{\mathcal{F}}})$ is used to subtract T^* since (T^*) contains a local obstruction at root. It requires, in advance, a generation of $\delta := \delta_m$ words in $\mathcal{L}_T$, denoted $(T_1, \dots, T_\delta)$. Each T_i is a word in the language $\mathcal{L}_{Tn_i}$, where $n_i := |T_i| < k$ and $\sum_{i=1}^{\delta} n_i = k - |m'|$. By the induction condition, T_i is a secondary structure in $\mathcal{T}_{n_i}$ for any i in $\{1, \dots, \delta\}$. In addition, the overlap-free condition implies that such m' is unique and there is no other local obstruction appears while constructing the word $m' \circ (T_1, \dots, T_\delta)$. Therefore, the word T^* subtracted once by the rule $T_{\bar{\mathcal{F}}} \to R_{m'}$.

This shows the correctness of the second rule and we can conclude that $\mathcal{T}_n = \mathcal{L}_{Tn}$.

In conclusion, Equation 6.1 holds for any value of n.

□

Next, we transfer the grammar into a system of functional equations

$$\begin{cases} S_{\bar{\mathcal{F}}}(z) &= z^2\, T_{\bar{\mathcal{F}}}(z)\, S_{\bar{\mathcal{F}}}(z) + z\, S_{\bar{\mathcal{F}}}(z) + 1 \\ T_{\bar{\mathcal{F}}}(z) &= S_{\bar{\mathcal{F}}}(z) - \sum_{m \in \mathcal{F}} z^{|m'|}\, T_{\bar{\mathcal{F}}}(z)^{\delta_m} \end{cases} \tag{6.2}$$

where $S_{\bar{\mathcal{F}}}(z)$ and $T_{\bar{\mathcal{F}}}(z)$ are, respectively, the OGF of the set $\mathcal{S}_{\bar{\mathcal{F}}}$ and $\mathcal{T}_{\bar{\mathcal{F}}}$.

6.2 COMPUTING AND ESTIMATING THE DOMINANT SINGULARITY

In general, Equation 6.2 can be solved using a symbolic calculus tool. However, this approach works poorly due to the sizeable local obstruction and the maximum open paired leaves number $\delta := \max_{m \in \mathcal{M}} \delta_m$. It requires solving an δ-degree equation of $T_{\bar{\mathcal{F}}}(z)$. Thus, we use an alternative approach, including dominant singularity computation.

From the second equation of Equation 6.2, one has the equality $S_{\bar{\mathcal{F}}}(z) = T_{\bar{\mathcal{F}}}(z) + \sum_{m \in \mathcal{F}} z^{|m'|}\, T_{\bar{\mathcal{F}}}(z)^{\delta_m}$. Replacing $S_{\bar{\mathcal{F}}}(z)$ in the first equation shows that $T_{\bar{\mathcal{F}}}(z)$ is a solution of $G(z, y) = y$ with

$$G(z, y) = z^2 y^2 + (z^2 R(z, y) + z)y + (z - 1)R(z, y) + 1$$

where $R(z,y) = \sum_{m\in\mathcal{F}} z^{|m'|}y^{\delta_m}$ is the OGF of the local obstruction set when $y = T_{\bar{\mathcal{F}}}(z)$. The polynomial $G(z,y)$ satisfies the conditions of Bender-Meir-Moon Theorem, which implies that

$$[z^n]\, T_{\bar{\mathcal{F}}}(z) := t_n = \Theta\left(\frac{\rho^{-n}}{n\sqrt{n}}\right)$$

with ρ is the dominant singularity of $T_{\bar{\mathcal{F}}}$, which is the non-zero root of the resultant of two polynomials in y,

$$P(z,y) = G(z,y) - y \qquad \text{and} \qquad Q(z,y) = \partial_y P(z,y).$$

Proposition 6.2: *The OGF $S_{\bar{\mathcal{F}}}(z)$ shares the same dominant singularity ρ as $T_{\bar{\mathcal{F}}}(z)$. Thus, coefficients of $S_{\bar{\mathcal{F}}}(z)$ satisfy*

$$[z^n]\, S_{\bar{\mathcal{F}}}(z) := s_n = \Theta\left(\frac{\rho^{-n}}{n\sqrt{n}}\right).$$

Proof. Let ρ be the dominant singularity of $T_{\bar{\mathcal{F}}}(z)$. Rewriting the first equation in Equation 6.2 gives $S_{\bar{\mathcal{F}}}(z) = 1/(1 - z - z^2 T_{\bar{\mathcal{F}}}(z))$. The denominator is non-zero, otherwise $T_{\bar{\mathcal{F}}}(z) = (1-z)/z^2$. Therefore, $S_{\bar{\mathcal{F}}}(z)$ shares the same dominant singularity with $T_{\bar{\mathcal{F}}}(z)$. □

Unfortunately, symbolic calculus tools, such as SymPy, still cannot support the resultant root finding in our case because of numerical instability issues. To workaround, we estimate the dominant singularity based on the exact coefficient of $S_{\bar{\mathcal{F}}}(z)$. Given a positive integer k, one has

$$\rho = \lim_{n\to\infty}\left(\frac{s_n}{s_{n-k}}\right)^{\frac{1}{k}}$$

where s_i is the i-th coefficient. In practice, $\hat{\rho}_n := (s_n/s_{n-k})^{1/k}$ is close enough to ρ for a large n. We use $n = 1\,000$ and $k = 20$ to estimate the dominant singularity in this study. In addition, this allows us to compute the factor $K_n := s_n/\hat{\rho}_n^{-n}$ for the asymptotic expression,

$$[z^n]\, S_{\bar{\mathcal{F}}}(z) \approx K_{1000}\frac{\hat{\rho}_{1000}^{-n}}{n\sqrt{n}}\left(1 + \mathcal{O}(1/n)\right). \tag{6.3}$$

It remains to us to determine the number of secondary structures of length n avoiding local obstructions, *i.e.*, s_n. The value can be computed with a dynamic programming algorithm based on recursive forms derived from the grammar in Proposition 6.1. One has,

$$s_n = \begin{cases} 1 & \text{if } n = 0 \\ s_{n-1} + \displaystyle\sum_{i=0}^{n-2} t_i^1 s_{n-i-2} & \text{otherwise} \end{cases}$$

where t_n^k with $l > 0$ is the cardinality of the set $\{(t_1, \cdots, t_k) \in \mathcal{T}^k; |t_1| + \cdots + |t_k| = n\}$ and $t_n^0 = 1$ for any n. For $k > 1$, the value of t_n^k is simply obtained via the recursive form below,

$$t_n^k = \sum_{i=0}^{n} t_{n-i}^1 t_i^{k-1}.$$

For $k = 1$, the second rule $T_{\mathcal{F}}(z) = S_{\mathcal{F}}(z) - \sum_{m \in \mathcal{M}} z^{|m'|} T_{\mathcal{F}}(z)^{\delta_m}$ implies that

$$t_n^1 = s_n - \sum_{m \in \mathcal{M}} t_{n-|m'|}^{\delta_m}.$$

EXAMPLE As a sanity test, we recompute the total number of secondary structures with a minimum distance $\theta = 3$. Indeed, structures with respect to the minimum distance are those avoiding the trivial motifs `()`, `(·)`, and `(··)`. The OGF of local obstruction set is then $R(z) = 1 + z + z^2$. Followed by the numerical procedure described above, the asymptotic number of secondary structures with a minimum distance $\theta = 3$ is

$$[z^n]\, S(z) := s_n = 0.71 \cdot \frac{2.289^n}{n\sqrt{n}} \left(1 + \mathcal{O}(1/n)\right). \tag{6.4}$$

These asymptotic values match the ones reported by Hofacker, Schuster, and Stadler [42].

6.3 UPPER-BOUND FOR DESIGNABLE SECONDARY STRUCTURES

Next, we applied the approach on different local obstruction sets $\mathcal{F}_{D \leqslant \varepsilon}$ obtained in Section 5.2.4 to have a first-order estimation for designable structure count (Equation 6.3). Dividing by Equation 6.4 yields an exponentially decreasing proportion of designable secondary structures, which we reported in Table 6.1. The base α of proportion is computed by

$$\alpha := \lim_{n \to \infty} \sqrt[n]{\frac{\#\text{Secondary structures avoiding } (\{(),(\bullet),(\bullet\bullet)\} \cup \mathcal{F}_{D \leqslant \varepsilon})}{\#\text{Secondary structures avoiding } \{(),(\bullet),(\bullet\bullet)\}}}.$$

6.3.1 *Designable structures in basic inverse folding and superoptimal structures*

First, we consider the basic inverse folding settings, corresponding to suboptimal defect D^S with threshold $\varepsilon = 0$. We have obtained an overlap-free set $\mathcal{F}_{D^S \leqslant 0}$ of 387 local obstructions. Proposition 5.2 shows that secondary structures containing one of these obstructions are undesignable. Computing the dominant singularity gives $\rho = 0.4$, which implies the following asymptotic upper bound for the designable structure count in this model

$$\left|\mathcal{D}_n^{D^S \leqslant 0}\right| \leqslant 0.67 \cdot \frac{2.242^n}{n\sqrt{n}} \left(1 + \mathcal{O}(1/n)\right). \tag{6.5}$$

			Upper bound	Proportion of designable structures (upper bound)					
Defect	ε	ρ	$\lvert\mathcal{D}_n\rvert$	α	P_{50}	P_{100}	P_{200}	P_{500}	P_{1000}
D^S	0	0.4461	$0.67\frac{2.242^n}{n\sqrt{n}}$	0.9795	$3.35\,10^{-1}$	$1.19\,10^{-1}$	$1.50\,10^{-2}$	$2.98\,10^{-5}$	$9.40\,10^{-10}$
D^S	-1	0.4503	$0.73\frac{2.221^n}{n\sqrt{n}}$	0.9702	$2.27\,10^{-1}$	$5.00\,10^{-2}$	$2.44\,10^{-3}$	$2.82\,10^{-7}$	$7.72\,10^{-14}$
D^P	0.5	0.4466	$0.66\frac{2.239^n}{n\sqrt{n}}$	0.9782	$3.10\,10^{-1}$	$1.03\,10^{-1}$	$1.14\,10^{-2}$	$1.53\,10^{-5}$	$2.49\,10^{-10}$
D^P	0.1	0.4521	$0.71\frac{2.212^n}{n\sqrt{n}}$	0.9663	$1.79\,10^{-1}$	$3.23\,10^{-2}$	$1.05\,10^{-3}$	$3.59\,10^{-8}$	$1.30\,10^{-15}$
D^P	0.01	0.4621	$0.63\frac{2.164^n}{n\sqrt{n}}$	0.9455	$5.37\,10^{-2}$	$3.26\,10^{-3}$	$1.20\,10^{-5}$	$5.94\,10^{-13}$	$3.99\,10^{-25}$
D^E	1	0.4472	$0.65\frac{2.236^n}{n\sqrt{n}}$	0.9771	$2.89\,10^{-1}$	$9.08\,10^{-2}$	$8.94\,10^{-3}$	$8.52\,10^{-6}$	$7.86\,10^{-11}$

Table 6.1: Impact of local obstructions of length up to 14 on the proportion of actually designable secondary structures.

The probability for a secondary structure of length n with a minimum distance at 3, taken uniformly at random, to be designable is upper-bounded by

$$P_n = 0.944 \cdot 0.9795^n.$$

One concludes that, while about 1/3 of the structures of length 50 can be designed, this proportion quickly drops to less than 1.5% for RNAs of length 200, and reaches infinitesimal proportions ($9.4 \cdot 10^{-10}$) for large RNAs of length 1 000.

Next, we turn to a harder design objective with the threshold set at -1. In other words, the target structure should achieve an energy distance to its first suboptimal at least 1 kcal.mol^{-1}. With the obtained overlap-free obstruction set $\mathcal{F}_{D^S \leqslant -1}$ consisting of 573 motifs, the dominant singularity is found at $\rho = 0.4503$, leading to an asymptotic upper bound

$$\left|\mathcal{D}_n^{D^S \leqslant -1}\right| \leqslant 0.73 \cdot \frac{2.221^n}{n\sqrt{n}} \left(1 + \mathcal{O}(1/n)\right).$$

This implies the proportion of designable structures with respect to ($D^S \leqslant -1$) bounded by

$$P_n = 1.028 \cdot 0.9702^n.$$

Unsurprisingly, the proportion decreases much faster under the harder design condition. It drops to 5%, which is half of the one for classic inverse folding, for RNAs of length 100 and reaches $7.7 \cdot 10^{-14}$ for large RNAs of length 1 000, 10^{-4} less than the designable structure proportion in the classic setting.

6.3.2 *Structures with large equilibrium probabilities*

Next, we aim to design secondary structure that is not only the MFE of a sequence, but also has a large equilibrium probability. We set the design objective to the probability

defect D^P, and investigate the impact of ε on the proportion of designable secondary structures. We consider three overlap-free obstruction sets, $\mathcal{F}_{D^P \leqslant 0.5}$, $\mathcal{F}_{D^P \leqslant 0.1}$, and $\mathcal{F}_{D^P \leqslant 0.01}$, for three thresholds $\varepsilon \in \{0.5, 0.1, 0.01\}$, associated with targeted Boltzmann probabilities for the motifs greater than 50%, 90% and 99% respectively.

Interestingly, the $\varepsilon = 50\%$ case induces a dominant singularity of 0.4466, leading to a slightly slower asymptotic growth

$$\left|\mathcal{D}_n^{D^P \leqslant 50\%}\right| \leqslant 0.66 \cdot \frac{2.239^n}{n\sqrt{n}} \left(1 + \mathcal{O}(1/n)\right)$$

than for classic inverse folding. This is not entirely unexpected, since our definition of a valid design requires the target structure to be the sole MFE for the sequence. Thus, secondary structures satisfying some probability defect conditions must also be solutions to the inverse folding problem. However, the observed divergence of the two singularities suggests that an exponentially small proportion (albeit with a growth factor very close to 1) of MFE designs have a Boltzmann probability greater than 50%.

For defect thresholds of 0.1 and 0.01 on the probability, the departure from the MFE design is much more pronounced, with respective singularities at $z = 0.4521$ and $z = 0.4621$ respectively, leading to upper bounds in

$$\left|\mathcal{D}_n^{D^P \leqslant 10\%}\right| \leqslant 0.71 \cdot \frac{2.212^n}{n\sqrt{n}} \left(1 + \mathcal{O}(1/n)\right) \text{ and } \left|\mathcal{D}_n^{D^P \leqslant 1\%}\right| \leqslant 0.63 \cdot \frac{2.164^n}{n\sqrt{n}} \left(1 + \mathcal{O}(1/n)\right).$$

We obtain proportions of designable structures respectively bounded by

$$P_n = 0.9663^n \text{ and } P_n = 0.887 \cdot 0.9455^n.$$

Those estimates support the notion of an extreme sparsity of designable structures in the folding space, with only four out of 10^{-8} (resp. six out of 10^{-13}) structures of length 500 being designable for $\varepsilon = 0.1$ (resp. $\varepsilon = 0.01$). These abysmal proportions are consistent with the popular belief, which can be rigorously proven in the homopolymers model [26], that the Boltzmann probability of the MFE structure decreases exponentially with the sequence length in a random, uniformly distributed, RNA sequence.

6.3.3 *Designing structures having small expected equilibrium distance to their target structure*

Last but not least, we consider a design objective where the target structure should be central at the thermodynamic equilibrium. To that purpose, we enforce that, within the Boltzmann distribution, the expected base pair distance to the target structure remains smaller than one base pair. This corresponds to using the ensemble defect D^E with threshold $\varepsilon = 1$. Proposition 5.3 ensures that secondary structure containing obstructions in the set $\mathcal{F}_{D^E \leqslant 1}$ is not designable.

We obtain a slightly larger dominant singularity at $z = 0.4472$ comparing to the one for classic inverse folding. It leads to an asymptotic equivalent in

$$\left|\mathcal{D}_n^{D^E \leqslant 1}\right| \leqslant 0.65 \cdot \frac{2.236^n}{n\sqrt{n}} \left(1 + \mathcal{O}(1/n)\right)$$

and the proportion of designable structures bounded by

$$P_n = 0.930 \cdot 0.9771^n.$$

This proportion decreases slightly faster with the structure length than in the classic setting. At most 30% of all structures of length 50 are designable, and the proportion drops to $7.9 \cdot 10^{-11}$ for structures of length 1000.

However, this design objective is very stringent and not very realistic. By contrast, the ensemble defect tolerance allowed by popular software, such as NUPack [96], typically grows linearly with n, matching the expectation of increased diversity within the Boltzmann ensemble of larger RNAs. This emphasizes a shortcoming of the univariate approach, as no single motif is likely to contribute large enough values to the ensemble defect to exceed more realistic tolerances. Indeed, the most defective motif, of length up to 14, only contributes an expected distance of 3.40 base pairs. Thus, for higher tolerances, no local obstruction will be returned in reasonable time. Furthermore, secondary structures may contain motifs with small defect value but have a cumulated ensemble defect exceeding the threshold. This motivates the bivariate approach described in the next chapter, which exploits the (super)additivity of ensemble defects induced by different occurrences of motifs to produced refined bounds.

7

BOUNDING THE ASYMPTOTIC DISTRIBUTION OF SUPERADDITIVE DEFECTS

We have pointed out some drawbacks of the univariate analysis at the end of the previous chapter. To address these issues, we are interested in estimating the distribution of structure ensemble defect for a given length n, then computing the proportion of designable structures. Unfortunately, a polynomial-time algorithm to compute structure defect does not exist. Otherwise, one can determine if a structure is designable by comparing its defect and the tolerance in polynomial time. Therefore, we aim to set up a lower bound $\hat{D}_S$ for structure defects with the help of motifs occurring in the structure.

MOTIF COLLECTION. Instead of only using local obstructions for a certain tolerance ε as in Chapter 6, we consider all motifs with lengths up to 14, each associated with its minimum possible ensemble defect. The set is obtained by modifying Algorithm 5.2 such that the ensemble defect is returned at each folding and taking the minimum value of defects associated with each motif gives. For a motif that is not the MFE conformation adapted by any sequence, its minimum possible ensemble defect is set at 1 nt. Indeed, any sequence has at least a 50% chance to fold into an alternative motif with a distance of at least two nucleotides. Furthermore, motifs with null defects are excluded from the set since their contribution to the structure defect is null. We obtained a collection of 10 280 motifs with defects ranging from $1.7 \cdot 10^{-5}$ to 3.4.

7.1 SUPERADDITIVE ENSEMBLE DEFECT

More formally, we are interested in the following problem.

Problem 5: Give an overlap-free set of motifs, each associated with its minimum possible local ensemble defect, and a target secondary structure S^*, the goal is to find a lower bound for the minimum possible structure defect of S^*, $\min_w D^E(w, S^*)$, in the function of the minimum possible local defect of non-overlapping motifs occurring in the target, $\{\min_{w_m} D^{L,E}(w_m, m);\ m \in S^*\}$.

Intuitively, one would expect that the minimum possible structure defect to be at least equal to the additive contribution of motifs occurring in the target structure,

$$\min_w D^E(w, S^*) \geqslant \sum_{m \in S^*} \min_{w_m} D^{L,E}(w_m, m). \tag{7.1}$$

As seen in Proposition 5.3, it is not trivial to show the increase of ensemble defect while releasing the constraint on the root and the open-paired leaves of a motif. Moreover, suppose that the target structure contains two motifs sharing one base pair. In that case, two motifs consecutively occur in the target. The lower bound for the structure defect is then the minimum value between 2 and the sum of two motif defects due to the existence of the alternative structure, in which positions for the shared base pair are unpaired.

We are going to show that Equation 7.1 is valid for the restricted motif set, in which motif minimum possible ensemble defect is bounded by 1.

Proposition 7.1 (Superadditivity): *Let $\mathcal{M}$ be an overlap-free set of motifs, each associated with a minimum possible ensemble defect upper-bounded by 1 and S^* be the secondary structure. We have,*

$$\min_{w} D^E(w, S^*) \geqslant \sum_{\substack{m \in S^* \\ m \in \mathcal{M}}} \min_{w_m} D^{L,E}(w_m, m).$$

Proof. Let k be the number of motifs of $\mathcal{M}$ occurring in the target structure S^*. We denote these k motifs by $m_1, \ldots, m_k$. Given a secondary structure $S \in \mathcal{S}_{|S^*|}$, we define φ_S the number of motifs, of which the root of at least one of open-paired leaves is unpaired in S. Given an integer $l \in [0, k]$, we define the set of secondary structures $\mathcal{P}_l$, a subset of $\mathcal{S}_{|S^*|}$, as $\{S \in \mathcal{S}_{|S^*|};\ \varphi_S = l\}$. This creates a partition $\{\mathcal{P}_l\}_{l \in [0,k]}$ for the structure set $\mathcal{S}_{|S^*|}$.

Let w^* be a sequence of $\{A, C, G, U\}^{|S^*|}$ such that S^* is the MFE conformation of w^*. Let $w^*_{[|m_i|]}$ be the assignment on the positions of motif m_i for $i \in [1, k]$. The ensemble defect $D^E(w^*, S^*)$ is then

$$D^E(w^*, S^*) = \sum_{S \in \mathcal{S}_{|S^*|}} \mathbb{P}(S \mid w^*) \mathrm{DPdist}(S, S^*) = \sum_{l=0}^{k} p_l \sum_{S \in \mathcal{P}_l} \mathbb{P}_l(S \mid w^*) \mathrm{DPdist}(S, S^*)$$

where $p_l := \sum_{S \in \mathcal{P}_l} \mathbb{P}(S \mid w^*)$ is the total probability of the set $\mathcal{P}_l$ and $\mathbb{P}_l(S \mid w^*)$ is the structure probability defined over $\mathcal{P}_l$. Since the sum of p_l is 1, we have

$$D^E(w^*, S^*) \geqslant \min_{l} \sum_{S \in \mathcal{P}_l} \mathbb{P}_l(S \mid w^*) \mathrm{DPdist}(S, S^*).$$

For $l = 0$, motifs $m_1, \ldots, m_k$ occur in all structures in $\mathcal{P}_0$. In other words, the root and the open-paired leaves of each motif are considered to be constrained. Thus, within the set $\mathcal{P}_0$, we can compute independently the ensemble defect over positions corresponding to each motif and obtain

$$\sum_{S \in \mathcal{P}_0} \mathbb{P}_l(S \mid w^*) \mathrm{DPdist}(S, S^*) \geqslant \sum_{i=1}^{k} D^{L,E}(w^*_{[|m_i|]}, m_i) \geqslant \sum_{i=1}^{k} \min_{w_{m_i}} D^{L,E}(w_{m_i}, m_i).$$

For $l \in [1, k]$, we further partition the set $\mathcal{P}_l$ based on motifs counted in φ_S. Without loss of generality, we consider the subset of structures, denoted by $\mathcal{P}'$, such that the root and the open-paired leaves of motifs $m_{l+1}, \ldots, m_k$ remain paired. It means that ensemble defect over these motifs can be computed independently. For motifs $m_1, \ldots, m_l$, at least one base pair of each is unpaired. It involves, in total, at least $\lceil l/2 \rceil$ base pairs since two motifs can share the same base pairs, *i.e.* the root of one motif is an open-paired leaf of the other. Therefore, the distance to the target is at least $2 \cdot \lceil l/2 \rceil$ for the region other than $m_{l+1}, \ldots, m_k$. We have, with $\mathbb{P}'(S \mid w^*)$ be the structure probability defined over $\mathcal{P}'$,

$$\begin{aligned}
\sum_{S \in \mathcal{P}'} \mathbb{P}'(S \mid w^*) \mathrm{DPdist}(S, S^*) &\geqslant 2\lceil \frac{l}{2} \rceil + \sum_{i=l+1}^{k} D^{L,E}(w^*_{[|m_i|]}, m_i) \\
&\geqslant 2\lceil \frac{l}{2} \rceil + \sum_{i=l+1}^{k} \min_{w_{m_i}} D^{L,E}(w_{m_i}, m_i) \\
&\geqslant \sum_{i=1}^{k} \min_{w_{m_i}} D^{L,E}(w_{m_i}, m_i)
\end{aligned}$$

since the minumum possible local defect is upper-bound by 1. The inequality is valid for other subsets in $\mathcal{P}_l$. Thus,

$$\sum_{S \in \mathcal{P}_l} \mathbb{P}_l(S \mid w^*) \mathrm{DPdist}(S, S^*) \geqslant \sum_{i=1}^{k} \min_{w_{m_i}} D^{L,E}(w_{m_i}, m_i).$$

In conclusion, the structure defect $D^E(w^*, S^*)$ is at least $\sum_{i=1}^{k} \min_{w_{m_i}} D^{L,E}(w_{m_i}, m_i)$ as for the minimum possible structure defect $\min_w D^E(w, S^*)$. □

7.2 BIVARIATE ANALYSIS OF ENSEMBLE DEFECT WITH MINIMUM DISTANCE $\theta = 0$

In order to estimate the structure defect distribution given a structure length, we consider the bivariate generating function

$$S(z, u) = \sum_{S \in \mathcal{S}} z^{|S|} u^{\hat{D}_S} = \sum s_{n,d} z^n u^d \tag{7.2}$$

where $\hat{D}_S$ is the defect lower bound for structure S and $s_{n,d}$ is the number of secondary structures of length n, marked by z, with the defect at d, marked by u. One could notice that $S(z, 1)$ is the OGF of RNA secondary structures with a minimum distance $\theta = 0$. Let $\mathcal{M}$ be an overlap-free set of motifs. We assume, in this section, that the minimum possible ensemble defect of each motif is less or equal to 1. Proposition 7.1 shows that Equation 7.2 can be expressed with motif defect,

$$S(z, u) = \sum_{S \in \mathcal{S}} z^{|S|} u^{\hat{D}_S} = \sum_{S \in \mathcal{S}} \left(z^{|S|} u^{\sum_{m \in S} D_m} \right)$$

Like the previous problem (Proposition 6.1), we first describe a grammar that generates all secondary structures and increases the structure defect by motif defect at each motif occurrence. Then, we transfer the grammar into a system of functional equations, including $S(z,u)$. We are going to show that the system satisfies the condition of Drmota–Lalley–Woods Theorem, which implies that structure defect, a random variable denoted by D_n, has a Gaussian limiting distribution with the expected defect value and the variance of D_n linear to n, $\lim_{n\to\infty} \mathbb{E}[D_n] = \mu n$ and $\lim_{n\to\infty} \mathbb{V}[D_n] = \sigma^2 n$ where μ and σ are constants.

7.2.1 *Grammar describing lower bound for structure defect*

Here we set up a grammar that generates secondary structure and spotlights each motif occurrence where the defect is added. The first rule, the non-terminal symbol S, builds the set of all secondary structures ($\theta = 0$) while structures closed by a base pair is noted by the non-terminal symbol T. The second rule, T, constructs the set of structures within a base pair divided into two parts. The first one, $T \to \sum_{m\in\mathcal{M}} m' \circ (T,\dots,T)$, where each open paired leaf of m' becomes the parent of a non-terminal symbol T. Recall that m' is the part of motif m within the first base pair. The rule constructs the set of secondary structures that feature the motif at the root completing with a base pair. The second part, $T \to \bar{S}$, subtracts the contributions of secondary structures of the first part. In other words, $\bar{S} = S - \sum_{m\in\mathcal{M}} m' \circ (T,\dots,T)$.

Proposition 7.2 (Bivariate defect-marking grammars and systems): *Let $\mathcal{M}$ be an overlap-free set of motif, each associated with a minimum possible defect bounded by 1. The grammar below forms all secondary structures with respect to the minimum distance $\theta = 0$ while highlighting occurrences of motifs in $\mathcal{M}$.*

$$S \to (\,T\,)\,S + \bullet\, S + \varepsilon$$
$$T \to \bar{S} + \sum_{m\in\mathcal{M}} m' \circ (T,\dots,T)$$

with $\bar{S} = S - \sum_{m\in\mathcal{M}} m' \circ (T,\dots,T)$ and m' is the part delimited by the root of the motif m. The defect is a parameter that is initially defined on individual production rules of the grammar, such that all rules have defect 0 except for those of the form $T \to m' \circ (T,\dots,T)$, which have defect D_m. Then, the bivariate generating functions $S(z,u) = \sum_{S\in\mathcal{S}} z^{|S|} u^{\hat{D}_S}$ and $T(z,u) = \sum_{S\in\mathcal{T}} z^{|S|} u^{\hat{D}_S}$ satisfy the following system of functional equations,

$$\begin{cases} S(z,u) = z^2 T(z,u) S(z,u) + zS(z,u) + 1 \\ T(z,u) = S(z,u) + \sum_{m\in\mathcal{M}} T(z,u)^{\delta_m} (u^{D_m} - 1) z^{|m'|} \end{cases} \tag{7.3}$$

Proof. The first part of proof is similar to the one for Proposition 6.1 showing that the language generated by the grammar is indeed the real set of secondary structure

wanted. Denote by $\mathcal{S}_{n,k}$ the real set of all secondary structures (not the generated ones) having length n and defect k, and by $\mathcal{T}_{n,k}$ the ones of length n which, if surrounded by an extra pair of parentheses, would have defect k. Denote by $\mathcal{L}_S$ (resp. $\mathcal{L}_T$) the language generated from S (resp. T), and by $\mathcal{L}_{S\,n,k}$ (resp. $\mathcal{L}_{T\,n,k}$) its restriction to secondary structures of length n and defect k. By construction, the defect is additively extended over concatenations such that, given two words w and w', having defect D_w and $D_{w'}$ respectively, their concatenation $w.w'$ has defect $D_{w.w'} = D_w + D_{w'}$.

Lemma 7.3: *For all n, k we have*

$$\mathcal{L}_{S\,n,k} = \mathcal{S}_{n,k} \text{ and } \mathcal{L}_{T\,n,k} = \mathcal{T}_{n,k}; \tag{7.4}$$

This property, which we prove below, immediately implies that

$$|\mathcal{L}_{S\,n,k}| = s_{n,k} := |\mathcal{S}_{n,k}| \qquad \text{and} \qquad |\mathcal{L}_{T\,n,k}| = t_{n,k} := |\mathcal{T}_{n,k}|$$

The system of bivariate functional equations can be obtained by a direct application of the symbolic method [20, 30]. More precisely, this requires the grammar to be non-ambiguous. It is easy to see the non-ambiguity of the first rule, since $S \to (T)S$ and $S \to \bullet S$ construct two disjoint sets of words starting with two different letters (and $\bullet$. For the non-terminal symbol T, it is clear that sets produced from $T \to \bar{S}$ and $T \to \sum_{m \in \mathcal{M}} m' \circ (T, \ldots, T)$ are disjoint. In addition, the overlap-free condition implies the disjointedness among sets generated from each rule $T \to m' \circ (T, \ldots, T)$, from which one obtains the non-ambiguity of the second rule.

Since the rule $T \to m' \circ (T, \ldots, T)$ has defect at D_m for each m in $\mathcal{M}$, the rule corresponds to the generating function $u^{D_m} z^{|m'|} T^{\delta_m}$. Although the explicit form of the rule $T \to \bar{S}$ is unknown, there is an alternative way to derive its generating function using the fact that $\bar{\mathcal{S}} = \mathcal{S} \setminus \bigcup_{m \in \mathcal{M}} m' \circ (\mathcal{T}, \ldots, \mathcal{T})$,

$$\bar{S}(z, u) = S(z, u) - \sum_{m \in \mathcal{M}} z^{|m'|} T^{\delta_m}$$

To sum up,

$$\begin{aligned}
T(z, u) &= \bar{S}(z, u) + \sum_{m \in \mathcal{M}} D_m z^{|m'|} T^{\delta_m} \\
&= \left(S(z, u) - \sum_{m \in \mathcal{M}} z^{|m'|} T^{\delta_m} \right) + \sum_{m \in \mathcal{M}} D_m z^{|m'|} T^{\delta_m} \\
T(z, u) &= S(z, u) + \sum_{m \in \mathcal{M}} T(z, u)^{\delta_m} (u^{D_m} - 1) z^{|m'|}
\end{aligned}$$

□

Proof of Lemma 7.3. First, Equation 7.4 holds for $n = 0$ and any value of k. Since the set of secondary structures of size 0 is reduced to the empty structure ε, having

defect $D_\varepsilon = 0$, one has $\mathcal{S}_{0,0} = \{\varepsilon\}$, and $\mathcal{S}_{0,k>0} = \varnothing$. Moreover, the only word of length 0 generated from S is ε, via the rule $S \to \varepsilon$. Thus, $\mathcal{L}_{S\,0,0} = \{\varepsilon\}$ and $\mathcal{L}_{S\,0,k>0} = \varnothing$, from which we conclude that

$$\forall k \in \mathbb{N},\ \mathcal{L}_{S\,0,k} = \mathcal{S}_{0,k}.$$

The case of T is very similar, but depends on the presence/absence, in the motif list $\mathcal{M}$, of the motif m_ε that pairs two consecutive bases. Namely, one has $\mathcal{T} = \{\varepsilon\}$, but associated with a defect $d := D_{m_\varepsilon}$ if $m_\varepsilon \in \mathcal{M}$, or $d := 0$ otherwise. It follows that $\mathcal{T}_{0,d} = \{\varepsilon\}$ and $\mathcal{T}_{0,k\neq d} = \varnothing$. As for the grammar, the only derivations likely to produce a word of length 0 are either $T \to \bar{S} \rightsquigarrow \varepsilon$ if $m_\varepsilon \notin \mathcal{M}$, or $T \to m'_\varepsilon \circ (T, \ldots, T) \rightsquigarrow \varepsilon$ if $m_\varepsilon \in \mathcal{M}$, associated with defects $d = 0$ and $d = D_{m_\varepsilon}$ respectively, from which we conclude that

$$\mathcal{L}_{T\,0,k} = \left\{ \begin{array}{ll} \{\varepsilon\} & \text{if } k = d \\ \varnothing & \text{otherwise} \end{array} \right\} = \mathcal{T}_{0,k}, \forall k \in \mathbb{N}.$$

Next, let p be a positive integer. We assume that Equation 7.4 holds for any $n < p$, and consider the sets of secondary structures $\mathcal{S}_{p,k}$, $\mathcal{T}_{p,k}$, $\mathcal{L}_{S\,p,k}$ and $\mathcal{L}_{T\,p,k}$.

Consider a secondary structure S^* of length p with defect k, *i.e.* $S^* \in \mathcal{S}_{p,k}$. The first base is either paired with a base at position l or unpaired. We are going to show that, in both cases, S^* is also a word generated from the grammar and in the set $\mathcal{L}_{S\,p,k}$.

- In the paired case, S^* is the form $(T')S'$ with $T' \in \mathcal{T}'_{l-2,k_{T'}}$ and $S' \in \mathcal{S}_{p-l,k_{S'}}$, where $k_{T'}$ and $k_{S'}$ are the defect of secondary structures (T') and S', respectively. Since motifs occurring in S^* are either entirely contained in (T'), or in S', one has $k_{T'} + k_{S'} = k$. By the induction condition, T' and S' could be generated from the grammar. Then, S^* could be produced as follow, $S \to (T)S \rightsquigarrow (T')S' = S^*$ with $T' \in \mathcal{L}_{T\,l-2,k_{T'}}$ and $S' \in \mathcal{L}_{S\,p-l,k_{S'}}$.

  Because of the additivity of defect on concatenation, $D_{S^*} = D_{(T').S'} = D_{(T')} + D_{S'} = k_{T'} + k_{S'} = k$. It is easy to see that $|S^*| = p$. Thus $S^* \in \mathcal{L}_{S\,p,k}$.

- In the unpaired case, S^* is the form $\bullet S''$. Since S'' contains the same motifs as S^*, the defects of S'' and S^* are equal. It implies that $S'' \in \mathcal{S}_{p-1,k}$ and $S'' \in \mathcal{L}_{S\,p-1,k}$ by the induction condition. Therefore, S^* is a word of $\mathcal{L}_{S\,p,k}$ that could be generated via $S \to \bullet S \rightsquigarrow \bullet S'' = S^*$. This proves the completeness of the first rule.

Let $S^* \in \mathcal{L}_{S\,p,k}$ be a word generated from S. We will prove the correctness by showing that S^* is a valid secondary structure in $\mathcal{S}_{p,k}$. Consider a word S^*, produced from the non-terminal symbol S via one of the follows,

- $S \to (T)S \rightsquigarrow (T')S' = S^*$, where $T' \in \mathcal{L}_{T\,p_{T'},k_{T'}}$ and $S' \in \mathcal{L}_{S\,p_{S'},k_{S'}}$ with $p_{T'} + p_{S'} = p - 2$ and $k_{T'} + k_{S'} = k$. By induction condition, T' and S' are valid secondary structures in, respectively, $\mathcal{T}_{p_{T'},k_{T'}}$ and $\mathcal{S}_{p_{S'},k_{S'}}$. $(T')S'$ is then a valid secondary structure in $\mathcal{S}_{p_{T'}+2,k_{T'}}$.

- $S \to \bullet S \rightsquigarrow \bullet S'' = S^*$ with $S'' \in \mathcal{L}_{Sp-1,k}$. S'' is a valid secondary structures in $\mathcal{S}_{p-1,k}$ by induction condition. Moreover, one may note that $\bullet$ is a secondary structure in $\mathcal{S}_{1,0}$.

Since the concatenation of two secondary structures is also a secondary structure, one could conclude that S^* generated from S, either in the form $(T').S'$ or $\bullet.S''$, is indeed a secondary structure in $\mathcal{S}_{p,k}$. We have shown that $\mathcal{S}_{p,k} = \mathcal{L}_{Sp,k}$ for any value of k.

Let T^* be a secondary structure in $\mathcal{T}_{p,k}$. Depending on the existence of inner motif at the root, T^* is a secondary structure either in the set $\bar{\mathcal{S}}$ or $\bigcup_{m \in \mathcal{M}} m' \circ (\mathcal{T}, \ldots, \mathcal{T})$.

- If none of inner motif occurs in T^* at root level, one has $T^* \in \bar{\mathcal{S}}_{p,k} \subseteq \mathcal{S}_{p,k}$, which implies that T^* is a word in $\mathcal{L}_{Sp,k}$ using the result above. Thus, T^* is also a word in $\mathcal{L}_{Tp,k}$ that could be generated via $T \to \bar{S} \to S \rightsquigarrow T^*$.

- On the other hand, let m' be the inner motif that occurs in T^* at the root, *i.e.* $T^* \in m' \circ ((\mathcal{T}), \ldots, (\mathcal{T}))$. There exist a δ-tuple, $(T_1, \ldots, T_\delta) \in \mathcal{T}^\delta$, with $\delta := \delta_m$ such that $T^* = m' \circ ((T_1), \ldots, (T_\delta))$. Let $n_i := |T_i|$ and $k_i := D_{(T_i)}$ for i in $\{1, \ldots, \delta\}$. It is easy to see that

  $$\sum_{i=1}^{\delta} n_i = \sum_{i=1}^{\delta} |T_i| = |T^*| - |m'| = p - |m'|.$$

  The overlap-free condition implies that no motif occurs in m', *i.e.* $D_m = 0$ and each motif occurring in T^* is entirely contained in one of $\{(T_1), \ldots, (T_\delta)\}$, from which we derive that

  $$\sum_{i=1}^{\delta} k_i = \sum_{i=1}^{\delta} D_{(T_i)} = D_m + \sum_{i=1}^{\delta} D_{(T_i)} = D_{T^*} = D_{(T^*)} - D_m = k - D_m.$$

  Since T_i is a secondary structure in $\mathcal{T}_{n_i,k_i}$ with $n_i < p$ and $k_i < k$, T_i could be produced from the non-terminal symbol T and is a word in $\mathcal{L}_{Tn_i,k_i}$ by the induction condition. This leads us to the follow conclusion, T^* is a word generated from T via $T \to m' \circ (T, \ldots, T) \rightsquigarrow m' \circ ((T_1), \ldots, (T_\delta)) = T^*$. The length of T^* equals to $|m'| + \sum_{i=0}^{\delta} n_i = p$. Furthermore, the rule $T \to m' \circ (T, \ldots, T)$ increases the defect by D_m, from which we could derive that $D_{T^*} = D_m + \sum_{i=0}^{\delta} k_i = k$ and then $T^* \in \mathcal{L}_{Tp,k}$. From this, we conclude that

  $$\forall T^* \in \mathcal{T}_{p,k},\ T^* \in \mathcal{L}_{Tp,k} \tag{7.5}$$

Conversely, let $T^* \in \mathcal{L}_{Tp,k}$ be a word produced from the non-terminal symbol T. The word T^* is generated using either the rule $T \to \bar{S}$ or $T \to m' \circ (T, \ldots, T)$, where $m \in \mathcal{M}$ is one of inner motifs.

- In the first case, T^* is a word of $\mathcal{L}_{\bar{S}p,k}$, which is a subset of $\mathcal{L}_{Sp,k}$. Thus, T^* is a secondary structure of size p and defect k since $\mathcal{L}_{Sp,k} = \mathcal{S}_{p,k}$. In addition, since none of inner motifs occur in T^* at root by the definition of $\bar{S}$, the defect of (T^*) is equal to the one of T^*, which means that $T^* \in \mathcal{T}_{p,k}$;

- Otherwise, the rule $T \to m' \circ (T, \ldots, T)$ is used to produce T^*. It requires, in advance, a generation of $\delta := \delta_m$ words in $\mathcal{L}_T$, denoted $(T_1, \ldots, T_\delta)$. Each T_i is a word in the language $\mathcal{L}_{T_{n_i,k_i}}$, where $n_i := |T_i| < p$ and $k_i := D_{(T_i)} < k$ with $\sum_{i=1}^{\delta} n_i = p - |m'|$. Since the rule $T \to m' \circ (T, \ldots, T)$ increases the defect by D_m, one obtains $\sum_{i=0}^{\delta} k_i = k - D_m$. By the induction condition, T_i is a secondary structure in $\mathcal{T}_{n_i,k_i}$ for any i in $\{1, \ldots, \delta\}$. Therefore, $T^* = m' \circ (T_1, \ldots, T_\delta)$, where each T_i is added respectively to each paired leaf of m', is also a valid secondary structure. It is easy to see that $|T^*| = |m'| + \sum_{i=1}^{\delta} n_i = p$. The overlap-free condition implies that motif in T^* occurs entirely in one of (T_i). Thus, $D_{T^*} = \sum_{i=1}^{\delta} k_i$. Since the inner motif m' occurs at the root level of T^*, the secondary structure (T^*) contains the motif at the root, from which one has $D_{(T^*)} = D_m + D_{T^*} = D_m + \sum_{i=1}^{\delta} k_i = k$, then $T^* \in \mathcal{T}_{p,k}$. We obtain

$$\forall T^* \in \mathcal{L}_{T_{p,k}},\ T^* \in \mathcal{T}_{p,k} \tag{7.6}$$

From Equation 7.5 and Equation 7.6, we have $\mathcal{T}_{p,k} = \mathcal{L}_{T_{p,k}}$ for any value of k.

In conclusion, Equation 7.4 holds for any value of n and k. □

7.2.2 *Mean and variance computation*

Equation 7.3 is strongly connected and aperiodic. Drmota–Lalley–Woods Theorem implies that structure ensemble defect D_n has a Gaussian limiting distribution with the expected defect value and the variance linear to n,

$$\lim_{n\to\infty} \mathbb{E}[D_n] = \mu\, n \qquad \text{and} \qquad \lim_{n\to\infty} \mathbb{V}[D_n] = \sigma^2\, n$$

where μ and σ^2 are two constants to determine. Let us start with a simple case where the motif set consists of one motif.

EXAMPLE. Considering the trivial motif $\mathcal{M} = \{()\}$ introduced by the minimum distance $\theta = 1$ in the *Nussinov* energy model, motif $()$ has an ensemble defect D^E at 1 restricted by the hypothesis. In this example, one can easily rewrite Equation 7.3 to

$$z^2 S(z,u)^2 - (1 - z + (1-u)z^2) S(z,u) + 1 = 0.$$

Resolving this second degree equation of $S(z,u)$ yields

$$S(z,u) = \frac{(1 - z + (1-u)z^2) - \sqrt{(1 + z + (1-u)z^2)(1 - 3z + (1-u)z^2)}}{2z^2}.$$

The dominant singularity in the function of u is the root of $1 - 3z + (1-u)z^2$,

$$\rho(u) = \begin{cases} (3 - \sqrt{5 + 4u})/(2(1-u)) & \text{if } u \neq 1 \\ 1/3 & \text{if } u = 1 \end{cases}.$$

We have also $\rho'(1) = 1/9$ and $\rho''(1) = 4/243$. Drmota–Lalley–Woods Theorem shows that constants μ and σ^2 can be expressed with $\rho(1)$, $\rho'(1)$ and $\rho''(1)$,

$$\mu = -\rho'(1)/\rho(1) = 1/9 \qquad \text{and} \qquad \sigma^2 = -\rho''(1)/\rho(1) + \mu + \mu^2 = 2/27.$$

Unfortunately, in general case, such method is not simpler than a direct computation due to high degree of $T(z,u)$, *i.e.* the maximum number of motif open paired leaves in the collections. By definition, the expected value is the total defect divided by the number of secondary structures,

$$\mathbb{E}[D_n] = \frac{\sum_{S \in \mathcal{S}_n} \hat{D}_S}{|\mathcal{S}_n|} = \frac{[z^n] \left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1}}{[z^n]\, S(z,1)}$$

and the variance

$$\begin{aligned}
\mathbb{V}[D_n] = \mathbb{E}[D_n^2] - \mathbb{E}[D_n]^2 &= \frac{\sum_{S \in \mathcal{S}_n} \hat{D}_S^2}{|\mathcal{S}_n|} - \mathbb{E}[D_n]^2 \\
&= \frac{[z^n] \left.\frac{\partial}{\partial u}(u \frac{\partial S(z,u)}{\partial u})\right|_{u=1}}{[z^n]\, S(z,1)} - \left(\frac{[z^n] \left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1}}{[z^n]\, S(z,1)} \right)^2 \\
\mathbb{V}[D_n] &= \frac{[z^n] \left.\frac{\partial}{\partial u}(u \frac{\partial S(z,u)}{\partial u})\right|_{u=1} \times [z^n]\, S(z,1) - \left([z^n] \left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} \right)^2}{([z^n]\, S(z,1))^2}.
\end{aligned}$$

Proposition 7.4 (Mean and variance of structure defect): *Let $\mathcal{M}$ be an overlap-free set of motif, each associated with a minimum possible defect bounded by 1. For a positive integer n, the distribution of the ensemble defect D_n across unconstrained ($\theta = 0$) uniform secondary structures of length n follows a Normal limiting distribution of parameters:*

$$\lim_{n\to\infty} \mathbb{E}[D_n] = \mu\, n \qquad \textit{and} \qquad \lim_{n\to\infty} \mathbb{V}[D_n] = \sigma^2\, n$$

where

$$\mu = \sum_{m \in \mathcal{M}} D_m \times 3^{-|m| + \delta_m}$$

and

$$\begin{aligned}
\sigma^2 = &\Big(\sum_{m \in \mathcal{M}} D_m \Delta \Big) - 2 \Big(\sum_{m \in \mathcal{M}} \Delta \Big) \Big(\sum_{m \in \mathcal{M}} |m| \Delta \Big) + \Big(\sum_{m \in \mathcal{M}} \Delta \Big)^2 \\
&+ 8 \Big(\sum_{m \in \mathcal{M}} \Delta \Big) \Big(\sum_{m \in \mathcal{M}} \delta_m \Delta \Big) - \frac{3}{2} \Big(\sum_{m \in \mathcal{M}} \delta_m \Delta \Big)^2
\end{aligned}$$

with $\Delta := D_m \times 3^{-|m| + \delta_m}$.

Proof.
Expected value $\mathbb{E}[D_n]$. From Lemma 7.5 and Lemma 7.6, we have

$$[z^n]\, S(z,1) = \frac{3\sqrt{3}}{2\sqrt{\pi}} 3^n n^{-\frac{3}{2}} - \frac{117\sqrt{3}}{32\sqrt{\pi}} 3^n n^{-\frac{5}{2}} + o(3^n n^{-\frac{5}{2}})$$

$$\begin{aligned}[z^n]\, \frac{\partial S(z,u)}{\partial u}\bigg|_{u=1} &= \frac{3\sqrt{3}}{2\sqrt{\pi}} \sum_{m\in\mathcal{M}} \Delta 3^n n^{-\frac{1}{2}} + \frac{3\sqrt{3}}{4\sqrt{\pi}} \sum_{m\in\mathcal{M}} |m|\Delta 3^n n^{-\frac{3}{2}} - \frac{69\sqrt{3}}{32\sqrt{\pi}} \sum_{m\in\mathcal{M}} \Delta 3^n n^{-\frac{3}{2}} \\ &\quad - \frac{3\sqrt{3}}{\sqrt{\pi}} \sum_{m\in\mathcal{M}} \delta_m \Delta 3^n n^{-\frac{3}{2}} - \frac{9\sqrt{3}}{8\sqrt{\pi}} \sum_{m\in\mathcal{M}} \delta_m^2 \Delta 3^n n^{-\frac{3}{2}} + o(3^n n^{-\frac{3}{2}}).\end{aligned}$$

Therefore,

$$\mathbb{E}[D_n] = \frac{[z^n]\, \frac{\partial S(z,u)}{\partial u}\Big|_{u=1}}{[z^n]\, S(z,1)} = \left(\sum_{m\in\mathcal{M}} D_m 3^{-|m|+\delta_m}\right) \times n + o(n).$$

Variance $\mathbb{V}[D_n]$. Recall that

$$\mathbb{V}[D_n] = \frac{[z^n]\, \frac{\partial}{\partial u}(u\frac{\partial S(z,u)}{\partial u})\Big|_{u=1} \times [z^n]\, S(z,1) - \left([z^n]\, \frac{\partial S(z,u)}{\partial u}\Big|_{u=1}\right)^2}{([z^n]\, S(z,1))^2}.$$

From the previous result, we have

$$([z^n]\, S(z,1))^2 = \frac{27}{4\pi} 3^{2n} n^{-3} + o(3^{2n} n^{-3})$$

$$\begin{aligned}\left([z^n]\, \frac{\partial S(z,u)}{\partial u}\Big|_{u=1}\right)^2 &= \frac{27}{4\pi}\left(\sum_{m\in\mathcal{M}} \Delta\right)^2 3^{2n} n^{-1} + \frac{27}{4\pi}\left(\sum_{m\in\mathcal{M}} \Delta\right)\left(\sum_{m\in\mathcal{M}} |m|\Delta\right) 3^{2n} n^{-2} \\ &\quad - \frac{621}{32\pi}\left(\sum_{m\in\mathcal{M}} \Delta\right)^2 3^{2n} n^{-2} - \frac{27}{\pi}\left(\sum_{m\in\mathcal{M}} \Delta\right)\left(\sum_{m\in\mathcal{M}} \delta_m\Delta\right) 3^{2n} n^{-2} \\ &\quad - \frac{81}{8\pi}\left(\sum_{m\in\mathcal{M}} \Delta\right)\left(\sum_{m\in\mathcal{M}} \delta_m^2\Delta\right) 3^{2n} n^{-2} + o(3^{2n} n^{-2}).\end{aligned}$$

In addition, from Lemma 7.7

$$\begin{aligned}&[z^n]\, \frac{\partial}{\partial u}\left(u\frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1} \\ =&\frac{3\sqrt{3}}{2\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}} \Delta\right)^2 3^n n^{\frac{1}{2}} + \frac{3\sqrt{3}}{2\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}} D_m\Delta\right) 3^n n^{-\frac{1}{2}} + \frac{27\sqrt{3}}{32\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}} \Delta\right)^2 3^n n^{-\frac{1}{2}} \\ &- \frac{3\sqrt{3}}{2\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}} \Delta\right)\left(\sum_{m\in\mathcal{M}} |m|\Delta\right) 3^n 3^n n^{-\frac{1}{2}} + \frac{6\sqrt{3}}{\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}} \Delta\right)\left(\sum_{m\in\mathcal{M}} \delta_m\Delta\right) 3^n n^{-\frac{1}{2}} \\ &- \frac{9\sqrt{3}}{4\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}} \Delta\right)\left(\sum_{m\in\mathcal{M}} \delta_m^2\Delta\right) 3^n n^{-\frac{1}{2}} - \frac{9\sqrt{3}}{4\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}} \delta_m\Delta\right)^2 3^n n^{-\frac{1}{2}} + o(3^n n^{-\frac{1}{2}}).\end{aligned}$$

Then,

$$\begin{aligned}&[z^n]\, \frac{\partial}{\partial u}\left(u\frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1} \times [z^n]\, S(z,1) \\ &= \frac{27}{4\pi}\left(\sum_{m\in\mathcal{M}} \Delta\right)^2 3^{2n} n^{-1} + \frac{27}{4\pi}\left(\sum_{m\in\mathcal{M}} D(m)\Delta^2\right) 3^{2n} n^{-2} - \frac{405}{32\pi}\left(\sum_{m\in\mathcal{M}} \Delta\right)^2 3^{2n} n^{-2} \\ &\quad - \frac{27}{4\pi}\left(\sum_{m\in\mathcal{M}} \Delta\right)\left(\sum_{m\in\mathcal{M}} |m|\Delta\right) 3^{2n} n^{-2} + \frac{27}{\pi}\left(\sum_{m\in\mathcal{M}} \Delta\right)\left(\sum_{m\in\mathcal{M}} \delta_m\Delta\right) 3^{2n} n^{-2}\end{aligned}$$

$$-\frac{81}{8\pi}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}\delta_m^2\Delta\Big)3^{2n}n^{-2}-\frac{81}{8\pi}\Big(\sum_{m\in\mathcal{M}}\delta_m\Delta\Big)^2 3^{2n}n^{-2}+o(3^{2n}n^{-2})$$

$$[z^n]\ \frac{\partial}{\partial u}\left(u\frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1}\times[z^n]\,S(z,1)-\left([z^n]\ \frac{\partial S(z,u)}{\partial u}\Big|_{u=1}\right)^2$$

$$=\frac{27}{4\pi}\Big(\sum_{m\in\mathcal{M}}D(m)\Delta\Big)3^{2n}n^{-2}-\frac{27}{2\pi}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}|m|\Delta\Big)3^{2n}n^{-2}+\frac{27}{4\pi}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)^2 3^{2n}n^{-2}$$

$$+\frac{54}{\pi}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}\delta_m\Delta\Big)3^{2n}n^{-2}-\frac{81}{8\pi}\Big(\sum_{m\in\mathcal{M}}\delta_m\Delta\Big)^2 3^{2n}n^{-2}+o(3^{2n}n^{-2})$$

with $\Delta := D_m 3^{-|m|+\delta_m}$. Finally, we have $\mathbb{V}[D_n] = \sigma^2 \times n + o(n)$ with

$$\sigma^2 = \Big(\sum_{m\in\mathcal{M}}D(m)\Delta\Big)-2\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}|m|\Delta\Big)+\Big(\sum_{m\in\mathcal{M}}\Delta\Big)^2$$
$$+8\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}\delta_m\Delta\Big)-\frac{3}{2}\Big(\sum_{m\in\mathcal{M}}\delta_m\Delta\Big)^2.$$

□

Lemma 7.5: *We have* $T(z,1) = S(z,1)$ *and*

$$S(z,1) = \frac{1-z-\sqrt{(1+z)(1-3z)}}{2z^2}$$
$$[z^n]\,S(z,1) = \frac{3\sqrt{3}}{2\sqrt{\pi}}3^n n^{-\frac{3}{2}} - \frac{117\sqrt{3}}{32\sqrt{\pi}}3^n n^{-\frac{5}{2}} + o(3^n n^{-\frac{5}{2}}).$$

Proof. Replacing u by 1 in the second equation of Equation 7.3 yields the first equality and in the first equation gives

$$S(z,1) = z^2 S(z,1)^2 + zS(z,1) + 1$$

$$z^2 S(z,1)^2 - (1-z)S(z,1) + 1 = 0.$$

Solving the quadratic equation gives two solutions

$$S^+(z) = \frac{1-z+\sqrt{(1+z)(1-3z)}}{2z^2} \quad \text{and} \quad S^-(z) = \frac{1-z-\sqrt{(1+z)(1-3z)}}{2z^2}.$$

Both solutions have the dominant singularity at $z = \rho = 1/3$. However, the value of $[z^n]\,S^+(z)$ is negative when n is odd which against the non-negative number of secondary structures. Thus, $S(z,1) = S^-(z)$. Singular expansion around $z = \rho = 1/3$ following by the application of Flajolet and Odlyzko Theorem gives

$$S(z,1) = 3 - 3\sqrt{3}(1-\frac{z}{\rho})^{\frac{1}{2}} - \frac{45}{8}\sqrt{3}(1-\frac{z}{\rho})^{\frac{3}{2}} + o((1-\frac{z}{\rho})^{\frac{3}{2}})$$
$$[z^n]\,S(z,1) = \frac{3\sqrt{3}}{\sqrt{\pi}}\left(\frac{1}{2}+\frac{3}{16n}+o(\frac{1}{n})\right)\times 3^n n^{-\frac{3}{2}} - \frac{45\sqrt{3}}{8\sqrt{\pi}}\left(\frac{3}{4}+o(1)\right)\times 3^n n^{-\frac{5}{2}} + o(3^n n^{-\frac{5}{2}})$$
$$= \frac{3\sqrt{3}}{2\sqrt{\pi}}3^n n^{-\frac{3}{2}} - \frac{117\sqrt{3}}{32\sqrt{\pi}}3^n n^{-\frac{5}{2}} + o(3^n n^{-\frac{5}{2}}).$$

□

Lemma 7.6:

$$[z^n]\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} = \frac{3\sqrt{3}}{2\sqrt{\pi}}\sum_{m\in\mathcal{M}}\Delta 3^n n^{-\frac{1}{2}} + \frac{3\sqrt{3}}{4\sqrt{\pi}}\sum_{m\in\mathcal{M}}|m|\Delta 3^n n^{-\frac{3}{2}} - \frac{69\sqrt{3}}{32\sqrt{\pi}}\sum_{m\in\mathcal{M}}\Delta 3^n n^{-\frac{3}{2}}$$
$$-\frac{3\sqrt{3}}{\sqrt{\pi}}\sum_{m\in\mathcal{M}}\delta_m\Delta 3^n n^{-\frac{3}{2}} - \frac{9\sqrt{3}}{8\sqrt{\pi}}\sum_{m\in\mathcal{M}}\delta_m^2\Delta 3^n n^{-\frac{3}{2}} + o(3^n n^{-\frac{3}{2}})$$

with $\Delta := D_m 3^{-|m|+\delta_m}$.

Proof. The partial derivative of Equation 7.3 with respect to u is

$$\begin{cases} \dfrac{\partial S(z,u)}{\partial u} &= z^2\dfrac{\partial T(z,u)}{\partial u}S(z,u) + z^2T(z,u)\dfrac{\partial S(z,u)}{\partial u} + z\dfrac{\partial S(z,u)}{\partial u} \\ \dfrac{\partial T(z,u)}{\partial u} &= \dfrac{\partial S(z,u)}{\partial u} + \displaystyle\sum_{m\in\mathcal{M}}\delta_m T(z,u)^{\delta_m-1}\dfrac{\partial T(z,u)}{\partial u}(u^{D(m)}-1)z^{|m'|} \\ & \quad + \displaystyle\sum_{m\in\mathcal{M}} D_m T(z,u)^{\delta_m}u^{D(m)-1}z^{|m'|}. \end{cases}$$

Replacing u by 1 gives

$$\begin{cases} \left.\dfrac{\partial S(z,u)}{\partial u}\right|_{u=1} = z^2S(z,1)\left.\dfrac{\partial T(z,u)}{\partial u}\right|_{u=1} + z^2S(z,1)\left.\dfrac{\partial S(z,u)}{\partial u}\right|_{u=1} + z\left.\dfrac{\partial S(z,u)}{\partial u}\right|_{u=1} \\ \left.\dfrac{\partial T(z,u)}{\partial u}\right|_{u=1} = \left.\dfrac{\partial S(z,u)}{\partial u}\right|_{u=1} + \displaystyle\sum_{m\in\mathcal{M}} D_m S(z,1)^{\delta_m}z^{|m'|}. \end{cases}$$

Then, we replace $\left.\dfrac{\partial T(z,u)}{\partial u}\right|_{u=1}$ in the first equation by the second one,

$$\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} = z^2S(z,1)\left(\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} + \sum_{m\in\mathcal{M}} D_m S(z,1)^{\delta_m}z^{|m'|}\right)$$
$$+ z^2S(z,1)\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} + z\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1}$$
$$(1-z-2z^2S(z,1))\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} = \sum_{m\in\mathcal{M}} D_m S(z,1)^{\delta_m+1}z^{|m|}.$$

From the Lemma 7.5, we have $1-z-2z^2S(z,1) = \sqrt{(1+z)(1-3z)}$ and then

$$\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} = \sum_{m\in\mathcal{M}} D_m\frac{z^{|m|}S(z,1)^{\delta_m+1}}{\sqrt{(1+z)(1-3z)}}$$
$$= \sum_{m\in\mathcal{M}} D_m\Big(\frac{z^{|m|}}{\sqrt{1+z}}(\frac{1-z}{2z^2})^{\delta_m+1}(1-3z)^{-\frac{1}{2}} - \frac{z^{|m|}(\delta_m+1)(1-z)^{\delta_m}}{(2z^2)^{\delta_m+1}}$$
$$+ \frac{\delta_m(\delta_m+1)z^{|m|}(1-z)^{\delta_m-1}\sqrt{1+z}}{2(2z^2)^{\delta_m+1}}(1-3z)^{\frac{1}{2}} + o((1-3z)^{\frac{1}{2}})\Big).$$

Expansion on the dominant singularity $z=\rho=1/3$ is

$$\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} = \sum_{m\in\mathcal{M}} D_m\Big(\Big(\frac{3\sqrt{3}}{2}3^{-|m|+\delta_m}(1-\frac{z}{\rho})^{-\frac{1}{2}} + (\frac{3\sqrt{3}}{16} + \frac{15\sqrt{3}(\delta_m+1)}{4}$$
$$-\frac{3\sqrt{3}|m|}{2})3^{-|m|+\delta_m}(1-\frac{z}{\rho})^{\frac{1}{2}} + o((1-\frac{z}{\rho})^{\frac{1}{2}})\Big)$$
$$-\frac{9(\delta_m+1)}{2}3^{-|m|+\delta_m}\Big(1+O((1-\frac{z}{\rho})^{-1})\Big)$$

$$
\begin{aligned}
&+\frac{9\sqrt{3}\delta_m(\delta_m+1)}{4}3^{-|m|+\delta_m}(1-\frac{z}{\rho})^{\frac{1}{2}}+o((1-\frac{z}{\rho})^{\frac{1}{2}})\Big)\\
&=\frac{3\sqrt{3}}{2}\sum_{m\in\mathcal{M}}\Delta(1-\frac{z}{\rho})^{-\frac{1}{2}}-\frac{9}{2}\sum_{m\in\mathcal{M}}(\delta_m+1)\Delta+\Big(-\frac{3\sqrt{3}}{2}\sum_{m\in\mathcal{M}}|m|\Delta\\
&+\frac{63\sqrt{3}}{16}\sum_{m\in\mathcal{M}}\Delta+6\sqrt{3}\sum_{m\in\mathcal{M}}\delta_m\Delta+\frac{9\sqrt{3}}{4}\sum_{m\in\mathcal{M}}\delta_m^2\Delta\Big)(1-\frac{z}{\rho})^{\frac{1}{2}}+o((1-\frac{z}{\rho})^{\frac{1}{2}}
\end{aligned}
$$

with $\Delta = D_m 3^{-|m|+\delta_m}$. The asymptotic equivalent of n-th coefficient is then, from Flajolet and Odlyzko Theorem,

$$
\begin{aligned}
[z^n]\ \frac{\partial S(z,u)}{\partial u}\Big|_{u=1} &= \frac{3\sqrt{3}}{2\sqrt{\pi}}\sum_{m\in\mathcal{M}}\Delta(1-\frac{1}{8n}+o(\frac{1}{n}))3^n n^{-\frac{1}{2}}-\frac{1}{\sqrt{\pi}}\Big(\frac{63\sqrt{3}}{16}\sum_{m\in\mathcal{M}}\Delta-\frac{3\sqrt{3}}{2}\sum_{m\in\mathcal{M}}|m|\Delta\\
&\quad+6\sqrt{3}\sum_{m\in\mathcal{M}}\delta_m\Delta+\frac{9\sqrt{3}}{4}\sum_{m\in\mathcal{M}}\delta_m^2\Delta\Big)(\frac{1}{2}+o(1))3^n n^{-\frac{3}{2}}+o(3^n n^{-\frac{3}{2}})\\
&=\frac{3\sqrt{3}}{2\sqrt{\pi}}\sum_{m\in\mathcal{M}}\Delta 3^n n^{-\frac{1}{2}}+\frac{3\sqrt{3}}{4\sqrt{\pi}}\sum_{m\in\mathcal{M}}|m|\Delta 3^n n^{-\frac{3}{2}}-\frac{69\sqrt{3}}{32\sqrt{\pi}}\sum_{m\in\mathcal{M}}\Delta 3^n n^{-\frac{3}{2}}\\
&\quad-\frac{3\sqrt{3}}{\sqrt{\pi}}\sum_{m\in\mathcal{M}}\delta_m\Delta 3^n n^{-\frac{3}{2}}-\frac{9\sqrt{3}}{8\sqrt{\pi}}\sum_{m\in\mathcal{M}}\delta_m^2\Delta 3^n n^{-\frac{3}{2}}+o(3^n n^{-\frac{3}{2}}).
\end{aligned}
$$

□

Lemma 7.7:

$$
\begin{aligned}
&[z^n]\ \frac{\partial}{\partial u}\left(u\frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1}\\
&=\frac{3\sqrt{3}}{2\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)^2 3^n n^{\frac{1}{2}}+\frac{3\sqrt{3}}{2\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}D_m\Delta\Big)3^n n^{-\frac{1}{2}}+\frac{27\sqrt{3}}{32\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)^2 3^n n^{-\frac{1}{2}}\\
&\quad-\frac{3\sqrt{3}}{2\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}|m|\Delta\Big)3^n 3^n n^{-\frac{1}{2}}+\frac{6\sqrt{3}}{\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}\delta_m\Delta\Big)3^n n^{-\frac{1}{2}}\\
&\quad-\frac{9\sqrt{3}}{4\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}\delta_m^2\Delta\Big)3^n n^{-\frac{1}{2}}-\frac{9\sqrt{3}}{4\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\delta_m\Delta\Big)^2 3^n n^{-\frac{1}{2}}+o(3^n n^{-\frac{1}{2}})
\end{aligned}
$$

with $\Delta := D_m 3^{-|m|+\delta_m}$.

Proof. Let us recall the first order partial derivatives of $S(z,u)$ and $T(z,u)$,

$$
\begin{cases}
\dfrac{\partial S(z,u)}{\partial u} &= z^2\dfrac{\partial T(z,u)}{\partial u}S(z,u)+z^2T(z,u)\dfrac{\partial S(z,u)}{\partial u}+z\dfrac{\partial S(z,u)}{\partial u}\\
\dfrac{\partial T(z,u)}{\partial u} &= \dfrac{\partial S(z,u)}{\partial u}+\sum_{m\in\mathcal{M}}\delta_m T(z,u)^{\delta_m-1}\dfrac{\partial T(z,u)}{\partial u}(u^{D(m)}-1)z^{|m'|}\\
&\quad+\sum_{m\in\mathcal{M}}D_m T(z,u)^{\delta_m}u^{D(m)-1}z^{|m'|}.
\end{cases}
$$

The partial derivative of $u\dfrac{\partial S(z,u)}{\partial u}$ is then

$$
\begin{aligned}
\frac{\partial}{\partial u}\left(u\frac{\partial S(z,u)}{\partial u}\right) &= \frac{\partial}{\partial u}\left(z^2u\frac{\partial T(z,u)}{\partial u}S(z,u)\right)+\frac{\partial}{\partial u}\left(z^2T(z,u)u\frac{\partial S(z,u)}{\partial u}\right)+\frac{\partial}{\partial u}\left(zu\frac{\partial S(z,u)}{\partial u}\right)\\
&= z^2\frac{\partial}{\partial u}\left(u\frac{\partial T(z,u)}{\partial u}\right)S(z,u)+z^2u\frac{\partial T(z,u)}{\partial u}\frac{\partial S(z,u)}{\partial u}
\end{aligned}
$$

$$+ z^2 \frac{\partial T(z,u)}{\partial u} u \frac{\partial S(z,u)}{\partial u} + z^2 T(z,u) \frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right) + z \frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right)$$

$$\begin{aligned}\frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1} &= z^2 S(z,1)\ \frac{\partial}{\partial u}\left(u \frac{\partial T(z,u)}{\partial u}\right)\Big|_{u=1} + 2z^2\ \frac{\partial T(z,u)}{\partial u}\Big|_{u=1}\ \frac{\partial S(z,u)}{\partial u}\Big|_{u=1} \\ &\quad + z^2 S(z,1)\ \frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1} + z\ \frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1}.\end{aligned}$$

Same for $u \dfrac{\partial T(z,u)}{\partial u}$,

$$\begin{aligned}\frac{\partial}{\partial u}\left(u \frac{\partial T(z,u)}{\partial u}\right) &= \frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right) + \frac{\partial}{\partial u}\Big(\sum_{m \in \mathcal{M}} \delta_m u T(z,u)^{\delta_m - 1} \frac{\partial T(z,u)}{\partial u}(u^{D(m)} - 1) z^{|m'|}\Big) \\ &\quad + \frac{\partial}{\partial u}\Big(u \sum_{m \in \mathcal{M}} D_m T(z,u)^{\delta_m} u^{D(m)-1} z^{|m'|}\Big) \\ &= \frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right) + \sum_{m \in \mathcal{M}} \delta_m \frac{\partial}{\partial u}\left(u T(z,u)^{\delta_m - 1} \frac{\partial T(z,u)}{\partial u}\right)(u^{D(m)} - 1) z^{|m'|} \\ &\quad + \sum_{m \in \mathcal{M}} D_m \delta_m u T(z,u)^{\delta_m - 1} \frac{\partial T(z,u)}{\partial u} u^{D(m)-1} z^{|m'|} \\ &\quad + \sum_{m \in \mathcal{M}} D_m \delta_m \frac{\partial T(z,u)}{\partial u} T(z,u)^{\delta_m - 1} u^{D(m)} z^{|m'|} \\ &\quad + \sum_{m \in \mathcal{M}} D_m^2 T(z,u)^{\delta_m} u^{D(m)-1} z^{|m}\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial u}\left(u \frac{\partial T(z,u)}{\partial u}\right)\Big|_{u=1} &= \frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1} + 2 \sum_{m \in \mathcal{M}} D_m \delta_m S(z,1)^{\delta_m - 1}\ \frac{\partial T(z,u)}{\partial u}\Big|_{u=1} z^{|m'|} \\ &\quad + \sum_{m \in \mathcal{M}} D_m^2 S(z,1)^{\delta_m} u^{D(m)-1} z^{|m}.\end{aligned}$$

We then rewrite the equation of $\dfrac{\partial}{\partial u}\left(u \dfrac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1}$ and note that $z^2 z^{|m'|} = z^{|m|}$ for any motif m,

$$\begin{aligned}\frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1} &= z^2 S(z,1)\ \frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1} + 2 \sum_{m \in \mathcal{M}} D_m \delta_m S(z,1)^{\delta_m}\ \frac{\partial T(z,u)}{\partial u}\Big|_{u=1} z^{|m|} \\ &\quad + \sum_{m \in \mathcal{M}} D_m^2 S(z,1)^{\delta_m + 1} z^{|m|} + 2z^2\ \frac{\partial T(z,u)}{\partial u}\Big|_{u=1}\ \frac{\partial S(z,u)}{\partial u}\Big|_{u=1} \\ &\quad + z^2 S(z,1)\ \frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1} + z\ \frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1}\end{aligned}$$

$$\begin{aligned}(1 - z - 2z^2 S(z,1))\ \frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1} &= 2 \sum_{m \in \mathcal{M}} D_m \delta_m S(z,1)^{\delta_m}\ \frac{\partial T(z,u)}{\partial u}\Big|_{u=1} z^{|m|} \\ &\quad + \sum_{m \in \mathcal{M}} D_m^2 S(z,1)^{\delta_m + 1} z^{|m|} \\ &\quad + 2z^2\ \frac{\partial T(z,u)}{\partial u}\Big|_{u=1}\ \frac{\partial S(z,u)}{\partial u}\Big|_{u=1}.\end{aligned}$$

Since $\dfrac{\partial T(z,u)}{\partial u}\Big|_{u=1} = \dfrac{\partial S(z,u)}{\partial u}\Big|_{u=1} + \sum_{m \in \mathcal{M}} D_m z^{|m'|} S(z,1)^{\delta_m}$ and $1 - z - 2z^2 S(z,1) = \sqrt{(1+z)(1-3z)}$, the last equality is then,

$$\sqrt{(1+z)(1-3z)}\ \frac{\partial}{\partial u}\left(u \frac{\partial S(z,u)}{\partial u}\right)\Big|_{u=1}$$

$$
\begin{aligned}
&= \sum_{m\in\mathcal{M}} D_m^2 S(z,1)^{\delta_m+1} z^{|m|} + 2z^2\left(\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} + \sum_{m\in\mathcal{M}} D_m z^{|m'|} S(z,1)^{\delta_m}\right)\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} \\
&\quad + 2\sum_{m\in\mathcal{M}} D_m \delta_m S(z,1)^{\delta_m} z^{|m|}\left(\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} + \sum_{m\in\mathcal{M}} D_m z^{|m'|} S(z,1)^{\delta_m}\right) \\
&= \sum_{m\in\mathcal{M}} D_m^2 S(z,1)^{\delta_m+1} z^{|m|} + 2z^2\left(\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1}\right)^2 \\
&\quad + 2\sum_{m\in\mathcal{M}} D_m(\delta_m+1) S(z,1)^{\delta_m} z^{|m|}\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} \\
&\quad + 2\left(\sum_{m\in\mathcal{M}} D_m \delta_m S(z,1)^{\delta_m} z^{|m|}\right)\left(\sum_{m\in\mathcal{M}} D_m z^{|m'|} S(z,1)^{\delta_m}\right).
\end{aligned}
$$

Let

$$
\begin{aligned}
A_1(z) &= \sum_{m\in\mathcal{M}} D_m^2 z^{|m|} S(z,1)^{\delta_m+1} \Big/ \sqrt{(1+z)(1-3z)} \\
A_2(z) &= \left(2z^2\left(\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1}\right)^2\right. \\
&\qquad \left. + 2\left(\sum_{m\in\mathcal{M}} D_m(\delta_m+1) S(z,1)^{\delta_m} z^{|m|}\right)\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1}\right) \Big/ \sqrt{(1+z)(1-3z)} \\
A_3(z) &= 2\left(\sum_{m\in\mathcal{M}} D_m \delta_m S(z,1)^{\delta_m} z^{|m|}\right)\left(\sum_{m\in\mathcal{M}} D_m z^{|m'|} S(z,1)^{\delta_m}\right) \Big/ \sqrt{(1+z)(1-3z)}.
\end{aligned}
$$

We have

$$
\begin{aligned}
\left.\frac{\partial}{\partial u}\left(u\frac{\partial S(z,u)}{\partial u}\right)\right|_{u=1} &= A_1(z) + A_2(z) + A_3(z) \\
[z^n]\left.\frac{\partial}{\partial u}\left(u\frac{\partial S(z,u)}{\partial u}\right)\right|_{u=1} &= [z^n]\,A_1(z) + [z^n]\,A_2(z) + [z^n]\,A_3(z).
\end{aligned}
$$

The asymptotic expression for the nth coefficient of each A_i is given by lemmata below (Lemma 7.8, Lemma 7.9, and Lemma 7.10). Adding them yields

$$
\begin{aligned}
&[z^n]\left.\frac{\partial}{\partial u}\left(u\frac{\partial S(z,u)}{\partial u}\right)\right|_{u=1} \\
&= \frac{3\sqrt{3}}{2\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}}\Delta\right)^2 3^n n^{\frac{1}{2}} + \frac{3\sqrt{3}}{2\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}} D_m\Delta\right)3^n n^{-\frac{1}{2}} + \frac{27\sqrt{3}}{32\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}}\Delta\right)^2 3^n n^{-\frac{1}{2}} \\
&\quad - \frac{3\sqrt{3}}{2\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}}\Delta\right)\left(\sum_{m\in\mathcal{M}}|m|\Delta\right)3^n n^{-\frac{1}{2}} + \frac{6\sqrt{3}}{\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}}\Delta\right)\left(\sum_{m\in\mathcal{M}}\delta_m\Delta\right)3^n n^{-\frac{1}{2}} \\
&\quad - \frac{9\sqrt{3}}{4\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}}\Delta\right)\left(\sum_{m\in\mathcal{M}}\delta_m^2\Delta\right)3^n n^{-\frac{1}{2}} - \frac{9\sqrt{3}}{4\sqrt{\pi}}\left(\sum_{m\in\mathcal{M}}\delta_m\Delta\right)^2 3^n n^{-\frac{1}{2}} + o(3^n n^{-\frac{1}{2}})
\end{aligned}
$$

with $\Delta = D_m 3^{-|m|+\delta_m}$. □

Lemma 7.8: *Let*

$$A_1(z) = \sum_{m\in\mathcal{M}} D(m)^2 z^{|m|} S(z,1)^{\delta_m+1} \Big/ \sqrt{(1+z)(1-3z)}\,.$$

Then,

$$[z^n]\,A_1(z) = \frac{3\sqrt{3}}{2\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}} D(m)\Delta\Big)3^n n^{-\frac{1}{2}} + o(3^n n^{-\frac{1}{2}})$$

with $\Delta := D_m 3^{-|m|+\delta_m}$.

Proof.

$$\begin{aligned}
A_1(z) &= \sum_{m\in\mathcal{M}} D(m)^2 \frac{z^{|m|}}{\sqrt{1+z}}\left((\frac{1-z}{2z^2})^{\delta_m+1} + o(1)\right)(1-3z)^{-\frac{1}{2}} \\
&= \Big(\sum_{m\in\mathcal{M}} D(m)^2 \frac{z^{|m|}}{\sqrt{1+z}}(\frac{1-z}{2z^2})^{\delta_m+1}\Big)(1-3z)^{-\frac{1}{2}} + o((1-3z)^{-\frac{1}{2}}) \\
&= \frac{3\sqrt{3}}{2}\Big(\sum_{m\in\mathcal{M}} D(m)^2 3^{-|m|+\delta_m}\Big)(1-\frac{z}{\rho})^{-\frac{1}{2}} + o((1-\frac{z}{\rho})^{-\frac{1}{2}})
\end{aligned}$$

$$[z^n]\,A_1(z) = \frac{3\sqrt{3}}{2\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}} D(m)\Delta\Big)3^n n^{-\frac{1}{2}} + o(3^n n^{-\frac{1}{2}}).$$

□

Lemma 7.9: *Let*

$$A_2(z) = \frac{2z^2\left(\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1}\right)^2 + 2\left(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)S(z,1)^{\delta_m} z^{|m|}\right)\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1}}{\sqrt{(1+z)(1-3z)}}.$$

We have

$$\begin{aligned}
[z^n]\,A_2(z) = {} & \frac{3\sqrt{3}}{2\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)^2 3^n n^{\frac{1}{2}} + \frac{27\sqrt{3}}{32\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)^2 3^n n^{-\frac{1}{2}} \\
& - \frac{3\sqrt{3}}{2\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}|m|\Delta\Big)3^n n^{-\frac{1}{2}} \\
& - \frac{3\sqrt{3}}{\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}\delta_m\Delta\Big)3^n n^{-\frac{1}{2}} \\
& - \frac{9\sqrt{3}}{4\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}\delta_m^2\Delta\Big)3^n n^{-\frac{1}{2}} \\
& - \frac{9\sqrt{3}}{4\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\delta_m\Delta\Big)^2 3^n n^{-\frac{1}{2}} + o(n^{-\frac{1}{2}})
\end{aligned}$$

with $\Delta := D_m 3^{-|m|+\delta_m}$.

Proof. Note that, calculated in Lemma 7.6,

$$\left.\frac{\partial S(z,u)}{\partial u}\right|_{u=1} = \frac{1}{(1+z)^{\frac{1}{2}}(1-3z)^{\frac{1}{2}}} \sum_{m\in\mathcal{M}} D_m z^{|m|} S(z,1)^{\delta_m+1}.$$

The first term of $A_2(z)$ is equal to

$$A_{21}(z) = \frac{2z^2}{(1+z)^{\frac{3}{2}}(1-3z)^{\frac{3}{2}}}\left(\sum_{m\in\mathcal{M}} D_m z^{|m|} S(z,1)^{\delta_m+1}\right)^2.$$

Development of $S(z,1)^{\delta_m+1}$ to the second order is needed to have a proper order of $(1-3z)$. From Lemma 7.5, we have

$$\begin{aligned} S(z,1)^{\delta_m+1} &= \left(\frac{1-z}{2z^2}\right)^{\delta_m+1} - \frac{(\delta_m+1)(1-z)^{\delta_m}\sqrt{(1+z)(1-3z)}}{(2z^2)^{\delta_m+1}} \\ &\quad + \frac{(\delta_m+1)\delta_m(1-z)^{\delta_m-1}(1+z)(1-3z)}{2(2z^2)^{\delta_m+1}} + o((1-3z)). \end{aligned}$$

$$\begin{aligned} A_{21}(z) &= \frac{2z^2}{(1+z)^{\frac{3}{2}}}\left(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m+1}\right)^2 (1-3z)^{-\frac{3}{2}} \\ &\quad - \frac{4z^2}{1+z}\left(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m+1}\right)\left(\sum_{m\in\mathcal{M}} D_m z^{|m|}\frac{(\delta_m+1)(1-z)^{\delta_m}}{(2z^2)^{\delta_m+1}}\right)(1-3z)^{-1} \\ &\quad + \frac{4z^2}{\sqrt{1+z}}\left(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m+1}\right) \\ &\qquad \left(\sum_{m\in\mathcal{M}} D_m z^{|m|}\frac{(\delta_m+1)\delta_m(1-z)^{\delta_m-1}}{2(2z^2)^{\delta_m+1}}\right)(1-3z)^{-\frac{1}{2}} \\ &\quad + \frac{2z^2}{\sqrt{1+z}}\left(\sum_{m\in\mathcal{M}} D_m z^{|m|}\frac{(\delta_m+1)(1-z)^{\delta_m}}{(2z^2)^{\delta_m+1}}\right)^2 \times (1-3z)^{-\frac{1}{2}} + o((1-3z)^{-\frac{1}{2}}) \\ &= B_1(z) + B_2(z) + B_3(z) + B_4(z) + o((1-3z)^{-\frac{1}{2}}) \end{aligned}$$

with

$$\begin{aligned} B_1(z) &= \frac{2z^2}{(1+z)^{\frac{3}{2}}}\left(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m+1}\right)^2 (1-3z)^{-\frac{3}{2}} \\ B_2(z) &= -\frac{4z^2}{1+z}\left(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m+1}\right)\left(\sum_{m\in\mathcal{M}} D_m z^{|m|}\frac{(\delta_m+1)(1-z)^{\delta_m}}{(2z^2)^{\delta_m+1}}\right)(1-3z)^{-1} \\ B_3(z) &= \frac{4z^2}{\sqrt{1+z}}\left(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m+1}\right) \\ &\qquad \left(\sum_{m\in\mathcal{M}} D_m z^{|m|}\frac{(\delta_m+1)\delta_m(1-z)^{\delta_m-1}}{2(2z^2)^{\delta_m+1}}\right)(1-3z)^{-\frac{1}{2}} \\ B_4(z) &= \frac{2z^2}{\sqrt{1+z}}\left(\sum_{m\in\mathcal{M}} D_m z^{|m|}\frac{(\delta_m+1)(1-z)^{\delta_m}}{(2z^2)^{\delta_m+1}}\right)^2 \times (1-3z)^{-\frac{1}{2}}. \end{aligned}$$

Similarly, the second term of $A_2(z)$ is

$$\begin{aligned} A_{22}(z) &= \frac{2}{(1+z)(1-3z)}\left(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)z^{|m|}S(z,1)^{\delta_m}\right)\left(\sum_{m\in\mathcal{M}} D_m z^{|m|}S(z,1)^{\delta_m+1}\right) \\ &= \frac{2}{(1+z)(1-3z)}\left(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)z^{|m|}\left((\frac{1-z}{2z^2})^{\delta_m}\right.\right. \\ &\quad \left.\left. - \frac{\delta_m(1-z)^{\delta_m-1}\sqrt{(1+z)(1-3z)}}{(2z^2)^{\delta_m}} + o((1-3z)^{\frac{1}{2}})\right)\right) \end{aligned}$$

$$\Big(\sum_{m\in\mathcal{M}} D_m z^{|m|}\Big((\frac{1-z}{2z^2})^{\delta_m+1} - \frac{(\delta_m+1)(1-z)^{\delta_m}\sqrt{(1+z)(1-3z)}}{(2z^2)^{\delta_m+1}} + o((1-3z)^{\frac{1}{2}})\Big)\Big)$$

$$\begin{aligned}
&= \frac{2}{1+z}\Big(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)z^{|m|}(\frac{1-z}{2z^2})^{\delta_m}\Big)\Big(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m+1}\Big)(1-3z)^{-1}\\
&\quad - \frac{2}{\sqrt{1+z}}\Big(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)z^{|m|}(\frac{1-z}{2z^2})^{\delta_m}\Big)\\
&\quad \Big(\sum_{m\in\mathcal{M}} D_m z^{|m|}\frac{(\delta_m+1)(1-z)^{\delta_m}}{(2z^2)^{\delta_m+1}}\Big)(1-3z)^{-\frac{1}{2}}\\
&\quad - \frac{2}{\sqrt{1+z}}\Big(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)z^{|m|}\frac{\delta_m(1-z)^{\delta_m-1}}{(2z^2)^{\delta_m}}\Big)\\
&\quad \Big(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m+1}\Big)(1-3z)^{-\frac{1}{2}} + o((1-3z)^{-\frac{1}{2}})\\
&= C_1(z) + C_2(z) + C_3(z) + o((1-3z)^{-\frac{1}{2}})
\end{aligned}$$

with

$$\begin{aligned}
C_1(z) &= \frac{2}{1+z}\Big(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)z^{|m|}(\frac{1-z}{2z^2})^{\delta_m}\Big)\Big(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m+1}\Big)(1-3z)^{-1}\\
C_2(z) &= -\frac{2}{\sqrt{1+z}}\Big(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)z^{|m|}(\frac{1-z}{2z^2})^{\delta_m}\Big)\\
&\quad \Big(\sum_{m\in\mathcal{M}} D_m z^{|m|}\frac{(\delta_m+1)(1-z)^{\delta_m}}{(2z^2)^{\delta_m+1}}\Big)(1-3z)^{-\frac{1}{2}}\\
C_3(z) &= -\frac{2}{\sqrt{1+z}}\Big(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)z^{|m|}\frac{\delta_m(1-z)^{\delta_m-1}}{(2z^2)^{\delta_m}}\Big)\\
&\quad \Big(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m+1}\Big)(1-3z)^{-\frac{1}{2}}.
\end{aligned}$$

One can notice that $B_2(z) + C_1(z) = 0$, $B_3(z) + C_3(z) = C_3(z)/2$, and $B_4(z) + C_2(z) = C_2(z)/2$. We have

$$\begin{aligned}
A_2(z) =& \frac{2z^2}{(1+z)^{\frac{3}{2}}}\Big(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m+1}\Big)^2 \times (1-3z)^{-\frac{3}{2}}\\
&- \frac{1}{\sqrt{1+z}}\Big(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m+1}\Big)\Big(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)\delta_m z^{|m|}\frac{(1-z)^{\delta_m-1}}{(2z^2)^{\delta_m}}\Big)\\
&\times (1-3z)^{-\frac{1}{2}}\\
&- \frac{1}{\sqrt{1+z}}\Big(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)z^{|m|}(\frac{1-z}{2z^2})^{\delta_m}\Big)\Big(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)z^{|m|}\frac{(1-z)^{\delta_m}}{(2z^2)^{\delta_m+1}}\Big)\\
&\times (1-3z)^{-\frac{1}{2}} + o((1-3z)^{-\frac{1}{2}}).
\end{aligned}$$

Furthermore, the singular expansion on $z = \rho = 1/3$ gives

$$\begin{aligned}
A_2(z) =& \frac{3\sqrt{3}}{4}\Big(\sum_{m\in\mathcal{M}} D_m 3^{-|m|+\delta_m}\Big)^2 \times (1-\frac{z}{\rho})^{-\frac{3}{2}} - \frac{39\sqrt{3}}{32}\Big(\sum_{m\in\mathcal{M}} D_m 3^{-|m|+\delta_m}\Big)^2 \times (1-\frac{z}{\rho})^{-\frac{1}{2}}\\
&- \frac{3\sqrt{3}}{2}\Big(\sum_{m\in\mathcal{M}} D_m 3^{-|m|+\delta_m}\Big)\Big(\sum_{m\in\mathcal{M}} D_m|m|3^{-|m|+\delta_m}\Big) \times (1-\frac{z}{\rho})^{-\frac{1}{2}}\\
&+ \frac{15\sqrt{3}}{4}\Big(\sum_{m\in\mathcal{M}} D_m 3^{-|m|+\delta_m}\Big)\Big(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)3^{-|m|+\delta_m}\Big) \times (1-\frac{z}{\rho})^{-\frac{1}{2}}
\end{aligned}$$

$$-\frac{9\sqrt{3}}{4}\Big(\sum_{m\in\mathcal{M}} D_m 3^{-|m|+\delta_m}\Big)\Big(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)\delta_m 3^{-|m|+\delta_m}\Big)\times(1-\frac{z}{\rho})^{-\frac{1}{2}}$$
$$-\frac{9\sqrt{3}}{4}\Big(\sum_{m\in\mathcal{M}} D_m(\delta_m+1)3^{-|m|+\delta_m}\Big)^2\times(1-\frac{z}{\rho})^{-\frac{1}{2}}+o((1-\frac{z}{\rho})^{-\frac{1}{2}}).$$

Let $\Delta := D(m)3^{-|m|+\delta_m}$. Applying the theorem Flajolet and Odlyzko Theorem on each term with $\alpha = 3/2$ or $\alpha = 1/2$ gives the asymptotic equivalence

$$\begin{aligned}[z^n]\,A_2(z) =& \frac{3\sqrt{3}}{2\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)^2 3^n n^{\frac{1}{2}} - \frac{21\sqrt{3}}{32\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)^2 3^n n^{-\frac{1}{2}}\\ &-\frac{3\sqrt{3}}{2\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}|m|\Delta\Big)3^n n^{-\frac{1}{2}}\\ &+\frac{15\sqrt{3}}{4\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}(\delta_m+1)\Delta\Big)3^n n^{-\frac{1}{2}}\\ &-\frac{9\sqrt{3}}{4\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}(\delta_m+1)\delta_m\Delta\Big)3^n n^{-\frac{1}{2}}\\ &-\frac{9\sqrt{3}}{4\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}(\delta_m+1)\Delta\Big)^2 3^n n^{-\frac{1}{2}}+o(3^n n^{-\frac{1}{2}})\end{aligned}$$

$$\begin{aligned}[z^n]\,A_2(z) =& \frac{3\sqrt{3}}{2\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)^2 3^n n^{\frac{1}{2}} + \frac{27\sqrt{3}}{32\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)^2 3^n n^{-\frac{1}{2}}\\ &-\frac{3\sqrt{3}}{2\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}|m|\Delta\Big)3^n n^{-\frac{1}{2}}\\ &-\frac{3\sqrt{3}}{\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}\delta_m\Delta\Big)3^n n^{-\frac{1}{2}} - \frac{9\sqrt{3}}{4\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\Delta\Big)\Big(\sum_{m\in\mathcal{M}}\delta_m^2\Delta\Big)3^n n^{-\frac{1}{2}}\\ &-\frac{9\sqrt{3}}{4\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\delta_m\Delta\Big)^2 3^n n^{-\frac{1}{2}}+o(n^{-\frac{1}{2}}).\end{aligned}$$

□

Lemma 7.10: *Let*

$$A_3(z) = 2\Big(\sum_{m\in\mathcal{M}} D_m\delta_m S(z,1)^{\delta_m} z^{|m|}\Big)\Big(\sum_{m\in\mathcal{M}} D_m z^{|m'|} S(z,1)^{\delta_m}\Big)\Big/\sqrt{(1+z)(1-3z)}.$$

We have

$$[z^n]\,A_3(z) = \frac{9\sqrt{3}}{\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}}\delta_m\Delta\Big)\Big(\sum_{m\in\mathcal{M}}\Delta\Big)3^n n^{-\frac{1}{2}}+o(3^n n^{-\frac{1}{2}})$$

with $\Delta := D_m 3^{-|m|+\delta_m}$.

Proof.

$$\begin{aligned}A_3(z) =& \frac{2}{\sqrt{(1+z)(1-3z)}}\Big(\sum_{m\in\mathcal{M}} D_m\delta_m z^{|m|}\Big((\frac{1-z}{2z^2})^{\delta_m}+o(1)\Big)\Big)\\ &\Big(\sum_{m\in\mathcal{M}} D_m z^{|m'|}\Big((\frac{1-z}{2z^2})^{\delta_m}+o(1)\Big)\Big)\end{aligned}$$

$$
\begin{aligned}
&= \frac{2}{z^2\sqrt{1+z}}\Big(\sum_{m\in\mathcal{M}} D_m\delta_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m}\Big)\Big(\sum_{m\in\mathcal{M}} D_m z^{|m|}(\frac{1-z}{2z^2})^{\delta_m}\Big)(1-3z)^{-\frac{1}{2}} \\
&\quad + o((1-3z)^{-\frac{1}{2}}) \\
&= 9\sqrt{3}\Big(\sum_{m\in\mathcal{M}} D_m\delta_m 3^{-|m|+\delta_m}\Big)\Big(\sum_{m\in\mathcal{M}} D_m 3^{-|m|+\delta_m}\Big)(1-\frac{z}{\rho})^{-\frac{1}{2}} + o((1-\frac{z}{\rho})^{-\frac{1}{2}}).
\end{aligned}
$$

$$
[z^n]\,A_3(z) = \frac{9\sqrt{3}}{\sqrt{\pi}}\Big(\sum_{m\in\mathcal{M}} \delta_m\Delta\Big)\Big(\sum_{m\in\mathcal{M}} \Delta\Big)3^n n^{-\frac{1}{2}} + o(3^n n^{-\frac{1}{2}})
$$

with $\Delta = D_m 3^{-|m|+\delta_m}$. □

Proposition 7.4 presents a closed-form expression for the expected value and the variance of structure defect. Applying on motif $()$ with ensemble defect at 1 gives $\mu = 3^{-2}$ and $\sigma^2 = 3^{-2} - 4\cdot 3^{-4} + 3^{-4} = 2/27$, which are same as the values obtained using Drmota–Lalley–Woods Theorem.

Given a lower bound $\hat{D^E}$ for structure ensemble defect D^E and a tolerance $\varepsilon > 0$. We consider in this study the distribution of $\hat{D^E}$ under the uniform distribution of secondary structures, so that, the proportion of designable structures $\mathcal{D}_n^{D^E\leqslant\varepsilon}$ of length n obeys

$$
\frac{|\mathcal{D}_n^{D^E\leqslant\varepsilon}|}{|\mathcal{S}_n|} \leqslant \mathbb{P}\left(\hat{D^E} \leqslant \varepsilon\right).
$$

Since $\hat{D^E}$ follows a normal limiting distribution of mean μn and standard deviation $\sigma\sqrt{n}$, one has

$$
\mathbb{P}\left(\hat{D^E} \leqslant \varepsilon\right) = \Phi(-x) = 1 - \Phi(x) = 1 - \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{t^2}{2}}\,dt
$$

where

$$
x = \frac{\mu n - \varepsilon}{\sigma\sqrt{n}}
$$

and

$$
\Phi(z) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} e^{-\frac{t^2}{2}}\,dt
$$

is the cumulative distribution function of the standard normal distribution. Using integration by parts gives an asymptotic expansion,

$$
\mathbb{P}\left(\hat{D^E} \leqslant \varepsilon\right) = \frac{\sqrt{2}\,e^{-\frac{x^2}{2}}}{x\sqrt{\pi}}\sum_{i=0}^{\infty}(-1)^i\frac{(2i-1)!!}{x^{2n}}
$$

where $(2i-1)!!$ is the double factorial of $2i-1$. This implies an exponentially decreasing proportion of designable structures for any constant, sub-linear threshold ε, or even for a linearly increased one $\varepsilon = \kappa.n$ with $\kappa < \mu$.

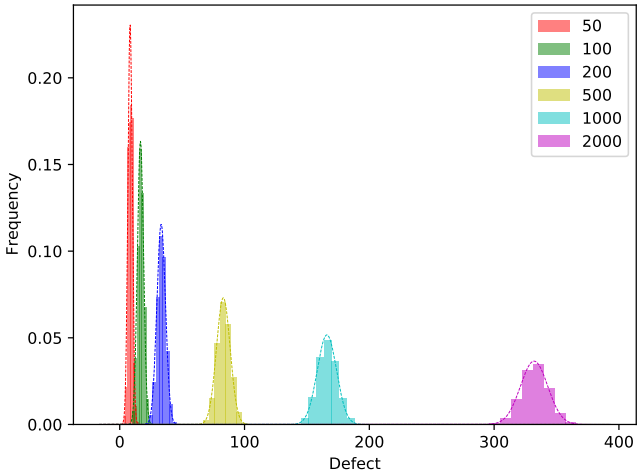


Figure 7.1: Empirical distribution of ensemble defect lower bound $\hat{D}^E$ with overlap-free motif over 10 000 uniformly sampled secondary structure with minimum distance $\theta = 0$. Curve in dashed style represents asymptotic distribution of $\hat{D}^E$ computed with Proposition 7.4.

7.2.3 *The distribution of $\hat{D}^E$ swiftly converges to its normal limiting distribution*

We have shown in Proposition 7.4 that structure ensemble defect follows a limiting normal distribution. However, this asymptotic approximation may represent poorly the defect distribution for smaller structures. In order to test the convergence, we compare the asymptotic and empirical distribution of this lower bound.

We add, in the collection of 10 280 motifs, three trivial motifs induced by the minimum distance $\theta = 3$, `()`, `(·)`, and `(··)`, with defect at 2 nts since the base pair can never be formed. To apply Proposition 7.4, we limit the defect value to 1 for motifs having defects larger than 1 nt, including trivial motifs. An overlap-free set of 808 motifs was then extracted using the approach described in Section 5.2.3. We prefer motif with higher defect per length in the second step to having a closed lower bound for structure defect. Applying Proposition 7.4 gives the values of two parameters $\mu = 0.1660$ and $\sigma^2 = 0.0596$. For instance, for a random uniformly distributed small RNA of length 50, the expected defect is 8.30 with the variance at 2.98.

On the other hand, the empirical distributions were estimated from a set of 10 000 secondary structures uniformly generated using GenRGenS [20] for length ranging from 50 to 2 000. Structure defect was computed by summing the defect of motifs occurring in the structure of interest. As shown in Figure 7.1, empirical distributions are virtually indistinguishable from their associated asymptotic distribution. In particular, the empirical distribution for structures of length 50 is estimated to have a mean at 8.66 and a variance at 3.07, comparing against 8.30 mean defect and 2.98 variance from limiting distribution. Moreover, for large RNA of length 2 000, the empirical distribution has estimated parameter $\mu = 0.1663$ and $\sigma^2 = 0.0589$, which coincide to the third digit with those of limiting distribution ($\mu = 0.1660$ and $\sigma^2 = 0.0596$).

7.3 NEW ENSEMBLE DEFECT LOWER BOUND WITH FULL MOTIF SET

Our approach in previous sections requires the motif set to be overlap-free, which restricts the size of the motif set. Indeed, we have reduced the set from 10 280 to 808 motifs. In order to use the complete motif set, we introduce a novel lower bound for structure ensemble defect defined as the maximum defect sum of non-overlapping motifs occurring in the structure,

$$\tilde{D}_S = \max_{\mathcal{P}} \sum_{m \in \mathcal{P}} D_m$$

where $\mathcal{P}$ is any set of motif occurrences in S such that any two occurrences do not share common positions. Given a secondary structure S, the new lower bound is computed using dynamic programming with naive pattern matching starting from the structure root. Let u be a node of S in tree presentation, *i.e.*, a base pair of S and $\mathrm{tr}(u)$ be the subtree of S starting from u. The new ensemble defect lower bound $\tilde{D}_{\mathrm{tr}(u)}$ for a subtree $\mathrm{tr}(u)$ is expressed in the following recursive form,

$$\tilde{D}_{\mathrm{tr}(u)} = \max \begin{cases} \displaystyle\sum_{c \text{ children of } u} \tilde{D}_{\mathrm{tr}(c)} \\ \displaystyle\max_{m \in \mathcal{M}_u} D_m + \sum_{i=i}^{\delta_m} \tilde{D}_{\mathrm{tr}(u_{m,i})} \end{cases}$$

where $\mathcal{M}_u$ is the set of motifs occurring at the root of $\mathrm{tr}(u)$, *i.e.* the node u, and $u_{m,i}$ is the node in subtree of u corresponding to the i-th open-paired leaf of motif m.

7.3.1 *Overlap-free motif set has a sufficient impact on structure defect estimation*

As in the previous section, we estimated the distribution of lower bound $\tilde{D}$ from a randomly and uniformly sampled set of 10 000 secondary structures $\theta = 0$ for different lengths $n \in \{50, 100, 200, 500, 1000, 2000\}$. As shown in Figure 7.2, the empirical distribution of this new lower bound $\tilde{D^E}$ is shifted to the right, compared to the asymptotic of $\hat{D^E}$. The expected defect per length μ increases from 0.1660 to 0.1868. This means that the 808 non-overlapping motifs selected in our analysis have a good enough impact on the ensemble defect of random secondary structure, and therefore on the combinatorics of designable structures.

7.3.2 *Ensemble defect lower bound with a minimum distance* $\theta = 3$

Among those motifs, the presence of trivial motifs (`()`, `(•)`, and `(••)`) is responsible for most of the defect, already inducing a value of 0.1605 for μ. To study the impact of non-trivial motifs, we imposed a minimal distance θ of 3 on secondary structures. This corresponds to consider trivial motifs as local obstructions and avoid them in

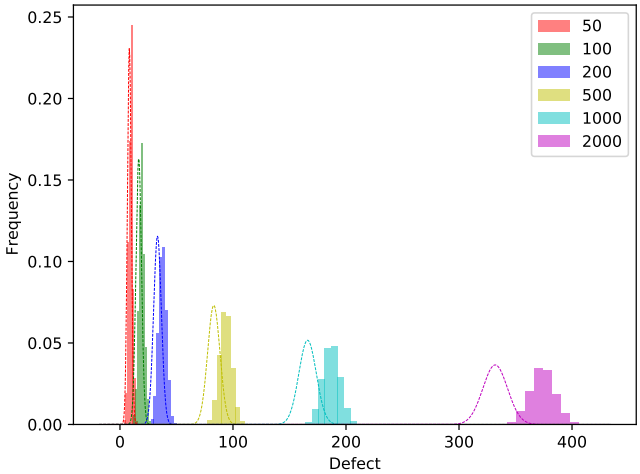


Figure 7.2: Empirical distribution of novel ensemble defect lower bound $\tilde{D}^E$ with overlap-free motif over 10 000 uniformly sampled secondary structure with minimum distance $\theta = 0$. Curve in dashed style represents asymptotic distribution of the former lower bound $\hat{D}^E$ computed with Proposition 7.4 using an overlap-free motif set.

the rule of non-terminal symbol T as in Proposition 6.1. The system of functional equations (Equation 7.3) becomes then

$$\begin{cases} S(z,u) = z^2 T(z,u) S(z,u) + zS(z,u) + 1 \\ T(z,u) = S(z,u) - 1 - z - z^2 + \sum_{m \in \mathcal{M}} T(z,u)^{\delta_m} (u^{D_m} - 1) z^{|m'|} \end{cases}.$$

Putting $u = 1$ gives a quadratic equation of $S(z,1)$,

$$z^2 S(z,1)^2 - (1 - z + z^2 + z^3 + z^4) S(z,1) + 1 = 0.$$

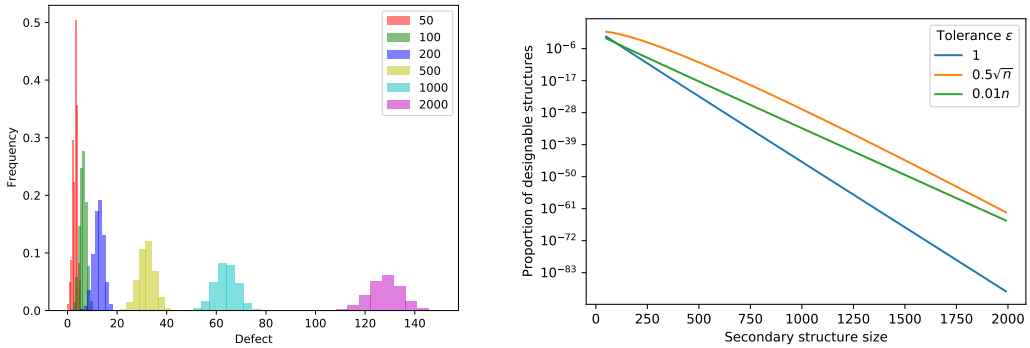
Solving it gives,

$$S(z,1) = \frac{1 - z + z^2 + z^3 + z^4 - \sqrt{(1 + z + z^2 + z^3 + z^4)(1 - 3z + z^2 + z^3 + z^4)}}{2z^2}.$$

Unfortunately, it becomes too complicated to compute the partial derivation of $S(z,u)$.

We estimated, as in Section 7.3, the empirical distribution of lower bound $\tilde{D}^E$ over 10 000 structures of different lengths and reported in Figure 7.3a. These empirical distributions suggest that the lower bound $\tilde{D}^E$ also follows a normal limiting distribution with mean and variance linear to the structure length. This is not overly surprising since the definition of the lower bound $\tilde{D}^E$ suggests, by nature, the existence of a system of functional equations satisfying the conditions of Drmota–Lalley–Woods Theorem [25]. In particular, the value of $\tilde{\mu}$ and $\tilde{\sigma}^2$ are, respectively, 0.064 and 0.02 for $n = 2000$, showing that a large proportion of the ensemble defect in Figure 7.2 can be attributed to the presence of trivial motifs.

We used these parameters to estimate in Figure 7.3b the proportion of designable secondary structures. The proportion can be seen to decrease exponentially with



(a) Empirical distribution

(b) Proportion of designable structures

Figure 7.3: Empirical distribution of ensemble defect lower bound $\tilde{D}^E$ using full set of motifs (a) and proportion of designable secondary structures (b) with minimal distance $\theta = 3$.

three different types of tolerance, constant, sub-linear, and linear. For a widely used tolerance in the popular design tool NUPack, $\varepsilon = 0.01n$, the proportion is around $1.3 \cdot 10^{-6}$ for a moderate length $n = 150$ and is less than 10^{-33} for large secondary structures of length 1000.

Part III

APPLICATION OF TREE DECOMPOSITIONS TO RNA DESIGN

8 INFRARED

We have shown in the previous chapters that designing a random secondary structure is challenging in theory. We are then, in this chapter, interested in finding proper RNA sequences. Hammer *et al.* presented a Fixed-Parameter Trackable (FPT) algorithm, RNARedPrint, to sample RNA sequences from a multivariate Boltzmann distribution for multiple target structures [37]. By adjusting the associated weights, sampled sequences can have specific free energy for each target and a certain GC content. In their work, the authors described a generic framework for multiple targets design with other design objectives, although it is not implemented in RNARedPrint.

In this chapter, we extend the framework for a more general Constraint Satisfaction Problem (CSP), and present InfraRed, an efficient and generic implementation of an Fixed-Parameter Trackable (FPT) algorithm. Section 8.1 defines the main problem and shows how a design problem is described as a CSP. We explain the mechanism of InfraRed in Section 8.2. As an usage example, we reimplement IncaRNAtion [69] in Section 8.3. In Section 8.4, we show that the framework is not only limited to RNA sequence sampling.

8.1 DESIGN PROBLEM AS CONSTRAINT SATISFACTION PROBLEM

A constraint network is composed of variables and a set of constraints imposed on variables. Given a constraint network, Constraint Satisfaction Problem (CSP) aims to find an assignment of variables that satisfies each constraint. Each variable is associated with a set of possible assignment values, named *domain*.

Definition 8.1 (Domain): The domain of a variable x, denoted by $D(x)$, is the set of possible assignments for x.

Example: *Given a target secondary structure* S *of length* n*, we consider a set of* n *variables* $X = \{x_1, \ldots, x_n\}$*, each represents one position in the target. A possible assignment for each variable* x_i *takes value from four nucleobases,* i.e. $D(x_i) = \{A, C, G, U\}$.

Definition 8.2 (Constraint): A constraint c defines assignments allowed for a set of variables $\{x_1, \ldots, x_l\}$. It is seen as a funciton taking an assignment $w_i \in D(x_i)$ for each variable x_i and returning a boolean value,

$$\begin{aligned} c: \quad & D(x_1) \times \cdots \times D(x_j) & \to \quad & \{\mathsf{True}, \mathsf{False}\} \\ & (w_1, \ldots, w_l) & \mapsto \quad & c(w_1, \ldots, w_l). \end{aligned}$$

The set $\{x_1, \ldots, x_l\}$ is called the *dependency* of constraint c, denoted by $\mathrm{dep}(c)$.

Example: *Given a secondary structure, one of common used constraints is BPComp, which forces the assignment of two positions can form a base pair. For each base pair (i, j) in the target structure, we define the constraint*

$$\mathit{BPComp}_{\{x_i, x_j\}}(w_i, w_j) = \begin{cases} \mathsf{True} & \text{if } (w_i, w_j) \in \mathcal{B} \\ \mathsf{False} & \text{otherwise} \end{cases}$$

with $\mathcal{B} = \{(\mathsf{A}, \mathsf{U}), (\mathsf{C}, \mathsf{G}), (\mathsf{G}, \mathsf{C}), (\mathsf{G}, \mathsf{U}), (\mathsf{U}, \mathsf{A}), (\mathsf{U}, \mathsf{G})\}$.

Definition 8.3 (Constraint Network): A *constraint network* $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ consists of

- A set of variables $\mathcal{X} = \{x_1, \ldots, x_n\}$;
- A set of domains of $\mathcal{X}$, $\mathcal{D} = D(x_1) \times \cdots \times D(x_n)$;
- A set of constraints $\mathcal{C} = \{c_1, \ldots, c_g\}$, each defined on a subset of $\mathcal{X}$.

In the classic Constraint Satisfaction Problem (CSP), the goal is to find an assignment for variables while respecting each constraint.

Problem 6 (Constraint Satisfaction Problem):
Input: Constraint network $(\mathcal{X}, \mathcal{D}, \mathcal{C})$
Output: Assignment w such that, for each constraint $c \in \mathcal{C}$, the assignment limited on the constraint dependency $w_{[|\mathrm{dep}(c)|]}$ is allowed, *i.e.* $c(w_{[|\mathrm{dep}(c)|]})$ is True.

The set of all solutions for the CSP, denoted by $\mathcal{D}_{\mathcal{C}}$, is a subset of $\mathcal{D}$.

Example (Naive Positive Design): *Given a secondary structure S of length n, we aim to determine RNA sequences that are compatible with the target structure as the first step in positive design. This is equivalent to the CSP with the constraint network $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ where*

- *Variables:* $\mathcal{X} = \{x_1, \ldots, x_n\}$;
- *Domain:* $\mathcal{D} = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{U}\}^n$;
- *Constraints: Constraint BPComp is defined for each base pair in the target structure.*

$$\mathcal{C} = \left\{ \mathit{BPComp}_{\{x_i, x_j\}};\ (i, j) \in S \right\}.$$

A solution for the CSP is a sequence compatible with the target structure.

Our framework InfraRed considers an extension of CSP, in which we introduce a novel set of weighted *functions* $\mathcal{F}$ in the constraint network. A function returns a value in real numbers given values for some variables. This gives the flexibility to capture assignment properties and allows to sample assignments according to their Boltzmann weight.

Definition 8.4 (Function): A function f is defined on a set of variables $\{x_1, \ldots, x_l\}$ with a weight $\beta_f \in \mathbb{R}$. It takes an assignment $w_i \in D(x_i)$ of each variable x_i and returns a value,

$$\begin{aligned} f: \quad & D(x_1) \times \cdots \times D(x_j) & \rightarrow \quad & \mathbb{R} \cup \{+\infty\} \\ & (w_1, \ldots, w_l) & \mapsto \quad & f(w_1, \ldots, w_l). \end{aligned}$$

Same as constraint, the set $\{x_1, \ldots, x_l\}$ is called the dependency of function f, denoted by $dep(f)$.

Example: *In a simplified energy model, we assume structure free energy is the sum of energies contributed from base pair stacks. Thus, the target structure energy can be captured by introducing energy function StackEnergy on each base pair stack (i, j) and $(i+1, j-1)$ in the target,*

$$\mathit{StackEnergy}_{\{x_i, x_j, x_{i+1}, x_{j-1}\}}(w_i, w_j, w_{i+1}, w_{j-1}) = \mathcal{E}_{stack}(w_i, w_j, w_{i+1}, w_{j-1})$$

with $\mathcal{E}_{stack}$ is the energy table for base pair stacks from Turner model [87].

The extended problem considered in the framework InfraRed is then,

Problem 7 (Generalized Design):
Input: Constraint network $(\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{F})$
Output: Assignment $w \in \mathcal{D}_{\mathcal{C}}$ compatible with constraints $\mathcal{C}$ such that

$$\mathbb{P}(w) \propto \prod_{\beta_f, f \in \mathcal{F}} e^{\beta_f \cdot f(w_{[|dep(f)|]})}.$$

This sampling problem can be seen as a subclass of valued CSP [74]. From now on, we simply say Problem 7 is a CSP for the readability reason. One can notice that constraint is a particular function with weight at 1, which returns only two values, 0 for True and $-\infty$ for False. Indeed, if an assignment dissatisfies one of the constraints, then the power of the base e is $-\infty$, meaning that its probability of being sampled is zero.

Example 1 (IncaRNAtion): *Here, we show how to describe a positive design problem as a CSP using IncaRNAtion [69] as an example. Given a target structure S of length n and a parameter γ, IncaRNAtion aims to sample a sequence w based on its GC-weighted Boltzmann probability*

$$\mathbb{P}(w) \propto e^{-\frac{\mathcal{E}(w,S)}{RT}} \gamma^{\#GC(w)}$$

where R *is the Boltzmann constant,* T *the temperature in Kelvin,* $\mathcal{E}$ *is structure energy using stacking energy model, and* $\#\mathsf{GC}(w) = \sum_{i=1}^{n} \mathrm{Id}_{w_i \in \{\mathsf{C},\mathsf{G}\}}$ *is the number of* G *or* C *in the sequence* w*.*
This is equivalent to the CSP *with the constraint network* $(\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{F})$ *with*

- *Variables:* $\mathcal{X} = \{x_1, \ldots, x_n\}$*;*
- *Domain:* $\mathcal{D} = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{U}\}^n$*;*
- *Constraints: Constraint* BPComp *is imposed on each base pair in the target structure,*

$$\mathcal{C} = \left\{ \mathit{BPComp}_{\{x_i, x_j\}};\ (i,j) \in S \right\};$$

- *Weighted Functions* $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$*:*
  Structure energy is modeled using the function $\mathit{StackEnergy}$*,*

$$\mathcal{F}_1 = \left\{ \left(-\frac{1}{\mathsf{RT}}, \mathit{StackEnergy}_{\{x_i, x_j, x_{i+1}, x_{j-1}\}} \right);\ (i,j) \in S \text{ if } (i+1, j-1) \in S \right\}.$$

  The GC*-content of an assignment* w *is captured by introducing the function* GCCont *on each variable* x_i*,*

$$\mathcal{F}_2 = \left\{ \left(\ln \gamma, \mathit{GCCont}_{\{x_i\}} \right);\ i \in \{1, \ldots, n\} \right\}$$

  with

$$\mathit{GCCont}_{\{x_i\}}(w_i) = \begin{cases} 1 & \text{if } w_i \in \{\mathsf{C}, \mathsf{G}\} \\ 0 & \text{otherwise.} \end{cases}$$

8.2 INFRARED CORE ENGINE

To solve Problem 7, one would need to compute the partition function,

$$Z_{\mathcal{X},\mathcal{D},\mathcal{C},\mathcal{F}} = \sum_{w \in \mathcal{D}_{\mathcal{C}}} \prod_{\beta_f, f \in \mathcal{F}} e^{\beta_f \cdot f(w_{[|\mathrm{dep}(f)|]})}. \tag{8.1}$$

In order to facilitate sequence sampling afterward, InfraRed precomputes the partition function using dynamic programming on a tree-like object derived from a constraint network. The approach is modified from the cluster tree elimination [19]. Stochastic backtracking is used to sample assignments on the same tree from the root to the leaves. Each function or constraint is evaluated once in one of the nodes or leaves to avoid redundancy during precomputation. Partial assignment dissatisfying a constraint is discarded. Therefore, the complexity is determined by the complexity of function/constraint evaluations in one node.

8.2.1 *Tree Decomposition*

First, we define *dependency graph*, a hypergraph induced from the dependencies of constraints and functions.

Definition 8.5 (Dependency graph): Given a constraint network $(\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{F})$, the dependency graph is defined as $G = (\mathcal{X}, \mathcal{E})$. Each vertex is a variable in the set $\mathcal{X}$ and each hyperedge in $\mathcal{E}$ is the dependency of a function or a constraint,

$$\mathcal{E} := \{\mathtt{dep}(c);\ c \in \mathcal{C}\} \cup \{\mathtt{dep}(f);\ f \in \mathcal{F}\}.$$

Next, we decompose dependency graph into a tree such that each node is a subset of $\mathcal{X}$.

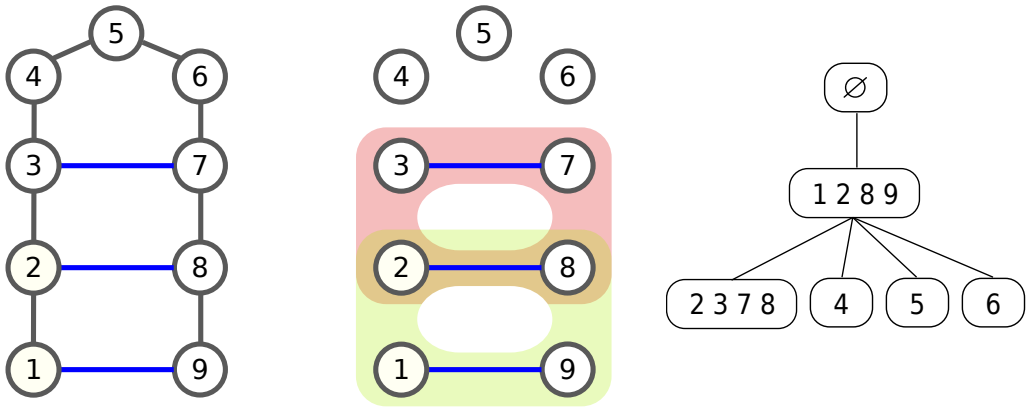
Definition 8.6 (Tree Decomposition): Given a dependency graph $G = (\mathcal{X}, \mathcal{E})$, a *tree decomposition* T of G is a tree (or forest) whose node u is a subset of $\mathcal{X}$, named *bag* and denoted by $\mathtt{bag}(u)$ such that

1. Each variable $x \in \mathcal{X}$ is in at least one bag;
2. For all hyperedge $e \in \mathcal{E}$, there is a node $u \in T$, such that $e \subseteq \mathtt{bag}(u)$;
3. For all variable $x \in \mathcal{X}$, the set $\{u \in T;\ x \in \mathtt{bag}(u)\}$ induces a connected tree.

A pseudo root with an empty bag is added to ensure the tree is connected. A random node of each connected component is selected to be a child of the pseudo root. Figure 8.1 shows an example of tree decomposition computed from a dependency graph. Tree decomposition captures necessary dependencies among variables from a given dependency graph. It assigns variables into different bags appropriately to divide the problem into several subproblems and resolve recursively. The first condition ensures that all variables are considered at least once while walking the tree T. Each function/constraint can be assigned to one node whose bag includes its dependency because of the second condition. The last one can also be stated that if k is a node in the path between two nodes u and v, then $\mathtt{bag}(u) \cap \mathtt{bag}(v) \subseteq \mathtt{bag}(k)$. It guarantees that the minimum needed dependency information can be passed from a node to another.

Definition 8.7 (Treewidth): The *width* of a tree decomposition is defined as $\max_{u \in T} |\mathtt{bag}(u)| - 1$. The *treewidth*, denoted by t_G, of a graph G is the minimum width among all possible tree decompositions.

Let T be a tree decomposition given a dependency graph G such that the width of T is t_G. Note that such a tree is not unique. The possible assignments in each bag are the Cartesian product of each variable domain in the bag. If variables have equal



(a) Target Structure (b) Dependency Graph (c) Tree Decomposition

Figure 8.1: (a) Target secondary structure to design $(((\cdots)))$. (b) Associated dependency graph (b) includes 3 hyperedges of 2 vertices (blue) introduced by the base pair complementary constraints and 2 hyperedges of 4 vertices (red and green) introduced by functions interpreting stack energy model. (c) A possible tree decomposition of width 3.

domain size d, the assignment amount is bounded by d^{t_G+1}. Since function evaluation generally takes a polynomial time, partition function computation's complexity is determined by the treewidth t_G. Although determining whether a dependency graph has treewidth at most a given value is an NP-complete problem [3], several heuristic approaches have been proposed to limit the treewidth, such as LibTW [21], libhtd [101], or min-fill-in provided by NetworkX [6] used in our framework.

8.2.2 *Partition Function Computation and Stochastic Backtracking*

Before partition function computation, we need to assign function in $\mathcal{F}$ and constraint in $\mathcal{C}$ to the tree. To avoid the redundancy, each is assigned to one node only such that its dependency is included in the bag. For simplicity reason, we treat each constraint $c \in \mathcal{C}$ as a weighted function with a weight at 1 that returns 1 if the given assignment satisfies the constraint and $-\infty$ otherwise. The difference in the implementation is explained in Section 8.3.

Definition 8.8 (Cluster Tree): Given a constraint network $(\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{F})$, a *cluster tree* is a tree decomposition T obtained from the dependency graph with a function/constraint assignment such that each function $f \in \mathcal{F}$ and constraint $c \in \mathcal{C}$ is assigned to an unique node $u \in T$ and $\mathrm{def}(f), \mathrm{def}(c) \subseteq \mathrm{bag}(u)$. We use $\zeta(u)$ to denote the set of functions/constraints assigned to the node u.

Let u be a node of a cluster tree T, and v be its parent. We define two variable sets,

- $\mathrm{sep}(u) := \mathrm{bag}(u) \cap \mathrm{bag}(v)$ the set of variables in common in two bags;

- $\mathtt{diff}(u) := \mathtt{bag}(u) \setminus \mathtt{sep}(u, v)$ the set of variables uniquely in the bag of child.

Furthermore, given a subset of variables $\mathcal{Y} := \{y_1, \ldots, y_h\} \subseteq \mathcal{X}$, we define the set of *partial assignments* as $\mathcal{D}(\mathcal{Y}) := \mathcal{D}(y_1) \times \cdots \times \mathcal{D}(y_h)$, the Cartesian product of variable domains. Let f be a function whose dependency is a subset of $\mathcal{Y}$, $\mathtt{dep}(f) \subseteq \mathcal{Y}$, and $w \in \mathcal{Y}$ be a partial assignment of $\mathcal{Y}$. The notation $f(w)$ refers to function evaluation on w limited to the dependency of f, *i.e.* $f(w) := f(w_{[|\mathtt{dep}(f)|]})$, for the readability reason.

Our algorithm travels the cluster tree in postorder while evaluating functions associated at each node. At node u, we aim to compute the partition function of the subtree $\mathsf{Tr}(u)$ of u for each partial assignment $w_1 \in \mathcal{D}(\mathtt{sep}(u))$ that is shared with the parent of u. In other words, the partition function is computed over the partial assignment set $\mathcal{D}_{w_1} := \{w \in \mathcal{D}(\mathtt{bag}(\mathsf{Tr}(u)));\ w_{[|\mathtt{sep}(u)|]} = w_1\}$ with $\mathtt{bag}(\mathsf{Tr}(u))$ is all variables in the subtree $\mathsf{Tr}(u)$,

$$Z_u(w_1) = \sum_{w \in \mathcal{D}_{w_1}} \prod_{\beta_f, f \in \zeta(\mathsf{Tr}(u))} e^{\beta_f \cdot f(w)}$$

which is equivalent to

$$Z_u(w_1) = \sum_{w_2 \in \mathcal{D}(\mathtt{bag}(\mathsf{Tr}(u)) \setminus \mathtt{sep}(u))} \prod_{\beta_f, f \in \zeta(\mathsf{Tr}(u))} e^{\beta_f \cdot f(w_1 \cup w_2)} \tag{8.2}$$

where $\zeta(\mathsf{Tr}(u))$ is the assigned functions in the subtree $\mathsf{Tr}(u)$. Summing Equation 8.2 over all partial assignment $w_1 \in \mathcal{D}(\mathtt{sep}(u))$ gives the total partition function of subtree of u.

Proposition 8.1: *Let $\{c_1, \ldots, c_k\}$ be children of node u. Equation 8.2 can be rewritten in a recursive form,*

$$Z_u(w_1) = \sum_{w_2 \in \mathcal{D}(\mathtt{diff}(u))} \left(\prod_{\beta_f, f \in \zeta(u)} e^{\beta_f \cdot f(w_1 \cup w_2)} \cdot \prod_{i=1}^{k} Z_{c_i}(w_1 \cup w_2) \right)$$

where $Z_{c_i}(w_1 \cup w_2) := (Z_{c_i}((w_1 \cup w_2)_{[|\mathtt{sep}(c_i)|]})$ is the partition function of the subtree of c_i for partial sequence in $\mathcal{D}(\mathtt{sep}(c_i))$.

Proof. Let $\{c_1, \ldots, c_k\}$ be children of node u. Equation 8.2 can be rewritten as

$$\begin{aligned} Z_u(w_1) &= \sum_{w_2 \in \mathcal{D}(\mathtt{bag}(\mathsf{Tr}(u)) \setminus \mathtt{sep}(u))} \left(\prod_{\beta_f, f \in \zeta(u)} e^{\beta_f \cdot f(w_1 \cup w_2)} \cdot \prod_{i=1}^{k} \prod_{\beta_f, f \in \zeta(\mathsf{Tr}(c_i))} e^{\beta_f \cdot f(w_1 \cup w_2)} \right) \\ &= \sum_{w_2 \in \mathcal{D}(\mathtt{diff}(u))} \left(\prod_{\beta_f, f \in \zeta(u)} e^{\beta_f \cdot f(w_1 \cup w_2)} \times \right. \\ &\qquad \left. \sum_{w_3 \in \mathcal{D}(\mathtt{bag}(\mathsf{Tr}(u)) \setminus \mathtt{bag}(u))} \prod_{i=1}^{k} \prod_{\beta_f, f \in \zeta(\mathsf{Tr}(c_i))} e^{\beta_f \cdot f(w_1 \cup w_2 \cup w_3)} \right) \end{aligned}$$

$$Z_u(w_1) \overset{*}{=} \sum_{w_2 \in \mathcal{D}(\mathtt{diff}(u))} \left(\prod_{\beta_f, f \in \zeta(u)} e^{\beta_f \cdot f(w_1 \cup w_2)} \times \prod_{i=1}^{k} \sum_{w_3 \in \mathcal{D}(\mathtt{bag}(\mathrm{Tr}(c_i)) \setminus \mathtt{sep}(c_i))} \prod_{\beta_f, f \in \zeta(\mathrm{Tr}(c_i))} e^{\beta_f \cdot f(w_1 \cup w_2 \cup w_3)} \right)$$

$$Z_u(w_1) \overset{**}{=} \sum_{w_2 \in \mathcal{D}(\mathtt{diff}(u))} \left(\prod_{\beta_f, f \in \zeta(u)} e^{\beta_f \cdot f(w_1 \cup w_2)} \cdot \prod_{i=1}^{k} Z_{c_i}(w_1 \cup w_2) \right)$$

The third condition of tree decomposition definition (see Definition 8.6) ensures the set $\{\mathtt{bag}(\mathrm{Tr}(c_i)) \setminus \mathtt{sep}(c_i)\}_{i \in [1,k]}$ is disjoint and $\mathtt{bag}(\mathrm{Tr}(u)) \setminus \mathtt{bag}(u) = \cup_{i=1}^{k} \mathtt{bag}(\mathrm{Tr}(c_i)) \setminus \mathtt{sep}(c_i)$, which valid the third equality $(*)$ above. The last equality $(**)$ sets up a relation between the partition function of the subtree of u and the ones of its children. Note that $Z_{c_i}(w_1 \cup w_2)$ is equivalent to $Z_{c_i}((w_1 \cup w_2)_{[|\mathtt{sep}(c_i)|]})$. □

Algorithm 8.1 shows the detail of partition function computation using the recursive form in Proposition 8.1. The total partition function Z is obtained at the root when the tree traversal ends.

Algorithm 8.1: Compute partition function given a cluster tree

Input : Cluster tree T
Output: $\mathcal{Z}_T := \{Z_u\}_u$ partition functions of each node u in T for all partial assignments $\mathcal{D}(\mathtt{sep}(u))$

```
Function PartitionFunction(T):
    Z_T ← ∅;
    forall node u of T in postorder do
        forall partial assignment w_1 ∈ D(sep(u)) do
            x ← 0;
            forall partial assignment w_2 ∈ D(diff(u)) do
                p ← product(e ^ (β_f · f(w_1 ∪ w_2)); β_f, f ∈ ζ(u))
                      · product(Z_c(w_1 ∪ w_2); child c of u);
                x ← x + p ;
            Z_u(w_1) ← x;
            Z_T ← Z_T ∪ {Z_u};
    return Z_T
```

With stochastic backtracking, one achieves sequence sampling from Boltzmann weighted distribution on cluster tree in the preorder tree traversal. At each node u, we add the assignment for $\mathtt{diff}(u)$ into the partial assignment. The preorder guarantees that variables in $\mathtt{sep}(u)$ are assigned before reaching node u. In other words, all variables $\mathtt{bag}(u)$ of node u are assigned to a value when tree traversal is at u. Sampling assignment for $\mathtt{diff}(u)$ requires the precomputed partition functions of children subtree. This is feasible since $\mathtt{sep}(c)$ is included in $\mathtt{bag}(u)$ for any child c of u. Algorithm 8.2 samples an assignment for all variables $\mathcal{X}$ from the Boltzmann-weighted distribution.

Algorithm 8.2: Stochastic backtracking for assignment sampling

Input : Cluster tree T, $\mathcal{Z}_T := \{Z_u\}_u$ partition functions of each node u in T for all partial assignments $\mathcal{D}(\mathrm{sep}(u))$
Output: w a random assignment sampled from Boltzmann weighted distribution
Function `AssignmentSampling(T, $\mathcal{Z}_T$)`:
  $w \leftarrow \varnothing$;
  forall *node* u *of* T *in preorder* **do**
    $x \leftarrow$ uniform random number between 0 and $Z_u(w)$;
    forall *partial assignment* $w_1 \in \mathcal{D}(\mathrm{diff}(u))$ **do**
      $p \leftarrow$ `product(`$e \hat{} \beta_f \cdot f(w \cup w_1)$; $\beta_f, f \in \zeta(u)$`)`
        $\cdot$ `product(`$Z_c(w \cup w_1)$; *child* c *of* u`)`;
      $x \leftarrow x - p$;
      if $x < 0$ **then**
        $w \leftarrow w \cup w_1$;
  return w

The time complexity of evaluating a function f is assumed to be polynomial to its dependency size $|\mathrm{dep}(f)|$. Indeed, most functions of interest are constant or linear in time. For example, it takes a constant time to access the energy table for energy functions. Let $\phi(n)$ be the maximum complexity among functions in $\mathcal{F}$ on an instance of size n. In addition, to limit the computation during backtracking, we force $\mathrm{diff}(u)$ to be a singleton for any node in the cluster tree. In other words, only one new variable to assign at each time in preorder. Such tree is easily obtained by inserting $\mathrm{diff}(u) - 1$ nodes between node u and its parent in the original cluster tree.

Proposition 8.2 (Complexity of InfraRed): *Let $(\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{F})$ be a constraint network, t be the treewidth of the associated dependency graph, and T be a cluster tree such that the width of T is t and $\mathrm{diff}(u) = 1$ for each node $u \in T$. The complexity of using InfraRed to generate k assignments with respect to Problem 7 is $\mathcal{O}((|\mathcal{X}| + |\mathcal{F} \cup \mathcal{C}|\phi(t))d^t + k|\mathcal{F} \cup \mathcal{C}|\phi(t)d)$ in time and $\mathcal{O}(|\mathcal{X}|d^s)$ in space with $s := \max_{u \in T} \mathrm{sep}(u)$ and $d := \max_{x \in \mathcal{X}} D(x)$ is the maximum variable domain size.*

Proof. At each node $u \in T$, Algorithm 8.1 evaluates at most $d^{|\mathrm{bag}(u)|}$ assignments and requires $d^{\mathrm{seq}(u)}$ in space to store the evaluation. For backtracking, Algorithm 8.2 computes $d^{|\mathrm{diff}(u)|}$ evaluations at each node. Since each function in $\mathcal{F}$ (or constraint in $\mathcal{C}$) is assigned to one and only one node, each function or constraint is called once in both algorithms. Evaluation takes in total $\mathcal{O}(|\mathcal{F} \cup \mathcal{C}|\phi(t)d^t)$ in time for Algorithm 8.1 and $\mathcal{O}(|\mathcal{F} \cup \mathcal{C}|\phi(t)d^{|\mathrm{diff}(u)|})$ for Algorithm 8.2. Under the assumption of $\mathrm{diff}(u) = 1$ for each node u, there are $|\mathcal{X}| + 1$ nodes in the cluster tree T. Thus, the total complexity to sample k assignments is $\mathcal{O}((|\mathcal{X}| + |\mathcal{F}|\phi(t))d^t + k|\mathcal{F}|\phi(t)d)$ in time and $\mathcal{O}(|\mathcal{X}|d^s)$ in space. □

Corollary 8.3: *InfraRed algorithm is in the class of Fixed-Parameter Trackable (FPT).*

8.2.3 *Multidimensional Boltzmann Sampling*

Recall the partition function of interest (Equation 8.1),

$$Z_{\mathcal{X},\mathcal{D},\mathcal{C},\mathcal{F}} = \sum_{w \in \mathcal{D}_{\mathcal{C}}} \prod_{\beta_f, f \in \mathcal{F}} e^{\beta_f \cdot f(w_{[|dep(f)|]})}.$$

The partition function can be seen as a multivariate generating function, where an assignment is a combinatorial object, and function f is the feature marked by the variable $\pi_f := e^{\beta_f}$. Therefore, the expected value for a function among all assignments, assuming other weights are fixed, is

$$\mathbb{E}[f] = \pi_f \frac{Z'(\pi_f)}{Z(\pi_f)}.$$

It means that sampled assignments can have a specific value for function f with a well-chosen weight. This can be extended to target specific values for different functions during sampling with a post-sampling rejection step, called multidimensional Boltzmann sampling [5]. InfraRed introduces a notion of feature contributed by several functions with the same weight for a more flexible application.

Definition 8.9 (Feature): Given a constraint network $(\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{F})$, a *feature* is a pair $(F, \mathcal{F}_F)$, where $F : \mathcal{D} \to \mathbb{R}$ is a function to evaluate an assignment of $\mathcal{D}$ and $\mathcal{F}_F \subseteq \mathcal{F}$ is a group of functions having the same weight.

The value of feature function F given an assignment w is usually, but not limited to, the total values of grouped functions, $F(w) = \sum_{f \in \mathcal{F}_F} f(w_{[|dep(f)|]})$. In some cases, a feature is too complicated to describe in the constraint network. The alternative is to decompose the feature into several simpler functions, each with a small dependency, and evaluate the feature's assignment after sampling. For example, a simpler energy model, such as the stacking energy model, is usually used in the framework to avoid large treewidth. The full Turner energy model is used in the post-sampling rejection step to target a specific structure free-energy.

InfraRed uses an iterative heuristic method to estimate proper feature weights. Let $F_1, \ldots, F_l$ be the features of interest and $\mu_1, \ldots, \mu_l$ be, respectively, the target values. Starting with initial weights $\beta_1^{[0]}, \ldots, \beta_l^{[0]}$, at each step t, InfraRed does

1. Generate assignment samples $\mathcal{A}$;
2. For i from 1 to l,
   a) Estimate the expected value of each feature F_i, $\hat{\mu}_i := \sum_{w \in \mathcal{A}} F_i(w)/|\mathcal{A}|$;

b) Update weights, $\beta_i^{[t]} = \beta_i^{[t-1]} + 0.01 * (\mu_i - \hat{\mu}_i)$;

c) Set new weight $\beta_i^{[t]}$ to the function group $\mathcal{F}_{F_i}$;

3. Accept assignment in $\mathcal{A}$ if the value to the target one is within a tolerance for each feature.

The iterations stops when enough good assignments are sampled.

8.3 IMPLEMENTATION AND USAGE

InfraRed is written in C++ and Python3. The source code is available at
https://gitlab.inria.fr/amibio/Infrared
and can be installed using conda
https://anaconda.org/conda-forge/infrared.

8.3.1 *Implementation*

InfraRed implementation consists of two parts, core engine in C++ and user interface in Python3. The framework is designed so that users only need to describe their design problem as a constraint network using the interface. The framework automatically manages the approach described in Section 8.2. The core engine includes the most time-consuming parts in the framework, partition function computation, and stochastic backtrack, while the interface takes care of multidimensional Boltzmann sampling. The code object-oriented with a generic type for flexibility, and maps classes between core and interface with pybind11.

FUNCTION Function and constraint are the two most essential components for formalizing the design problem. As seen in their definitions (see Definitions 8.2 and 8.4), the construction requires both a dependency and an evolution function for assignment. Thus, in the core engine, a function is implemented as a virtual class Function with a generic type for function value while assuming the default type is double. Class Function is constructed given a variable list as its dependency, which is implemented as the class Dependency, and its evaluation is a virtual function operator for users to define.

```
template<class FunValue=double> class Function : public Dependency {
public:
    using fun_value_t = FunValue;
    ...
    explicit
    Function(const std::vector<var_idx_t> &vars) : Dependency(vars) {}

    virtual fun_value_t operator () (const assignment_t &) const = 0;
    ...
}
```

Constraint is then the class Function with boolean.

```
using Constraint = Function<bool>;
```

While computing the partition function, constraints and functions are separately evaluated at each node. The constraint satisfaction is then an essential boolean operation checked at each node during the stochastic backtrack for the current partial assignment. The partition function of the partial assignment is added only if it passes the constraint satisfaction (Algorithm 8.2, second for loop). Furthermore, different functions/constraints can have different dependencies but share the same assignment evaluation function. A materialization is implemented to store the result after the first assignment evaluation to avoid redundant computation.

8.3.2 *InfraRed Usage*

An interface of InfraRed in Python3 is offered to users to describe their design problem as Constraint Satisfaction Problem (CSP). A reimplementation of IncaRNAtion [69] is presented below to illustrate the usage of the framework. First, we start with importing the package and declaring the target structure.

```
# incarnation.py
import infrared as ir

target = "((((...)))).(((...)))"
seqlen = len(target)
bps = ir.rna.parse(target) # list of target base pairs
```

Then, we initialize the model for a constraint network with one variable per position. The domain size of each variable is 4 with 0 for A, 1 for C, 2 for G, and 3 for U. One can omit the variable name X in case of unambiguity.

```
model = ir.Model()
model.add_variables(seqlen, 4, 'X')
idx = model.idx # function to get named variable index
Xidx = idx([('X',i) for i in range(seqlen)]) # indices of positional
    variables
```

Users-defined function/constraint is the class inherited from Function/Constraint. In order to avoid code redundancy, we offer in the interface a Python function to create a class, which takes two anonymous functions as input for dependency and evaluation. As seen in Example 1, we need constraint BPComp, function StackEnergy, and function GCCont to describe the design problem. Despite these functions and constraints are provided in the interface, we define them below as a demonstration.

```
# These are provided by InfraRed
_compltab = [(0,3),(1,2),(2,1),(2,3),(3,0),(3,2)] # base pair
    nucleotides

ir.def_constraint_class( 'ComplConstraint', lambda i,j: idx([('X',i), (
    'X',j)]), lambda x,y: (x,y) in _compltab)
```

```
ir.def_function_class( 'GCControl', lambda i: idx([('X',i)]), lambda x:
    1 if x==1 or x==2 else 0)
ir.def_function_class( 'StackEnergy', lambda i,j: idx([('X',i), ('X',j)
    , ('X',i+1), ('X',j-1)]), lambda x,y,x1,y1: ir.rna._stackenergy(x,y,
    x1,y1))
```

Next, we impose constraint BPComp on base pairs, add function StackEnergy on base pair stacks, and function GCCont on each position. Functions are grouped according to the given group name.

```
# Add Constraints and grouped Functions in constraint network
model.add_constraints(ComplConstraint(i, j) for (i, j) in bps)
model.add_functions([StackEnergy(i, j) for (i, j) in bps if (i+1, j-1)
    in bps], group='stackenergy')
model.add_functions([GCControl(i) for i in range(seqlen)], group='gc')
```

So far, a simple constraint network is defined. We can then construct a sampler to sample sequences based on Boltzmann-weighted distribution.

```
# A pretty printer
def print_sample(sample):
    seq = ir.rna.values_to_seq(sample.values())
    print("{} GC={:.2f}".format(seq, (seq.count('G')+seq.count('C'))/
        seqlen))

sampler = ir.BoltzmannSampler(model)
for i in range(5):
    sample = sampler.sample()
    print_sample(sample)
```

```
UAUAGGCUAUAUGGAGGUUCU GC=0.38
GGUCUGGGGUCCUUGAGGUGG GC=0.67
UGGUUGAGCCAAUGUUAGACG GC=0.48
GCCUUACAGGUGCGGGGGUUG GC=0.67
UGUUCGUGGUGUGCUUCAGGU GC=0.52
```

As seen above, GC content of sampled sequences varies from 38% to 67%, which suggests a good sequence diversity generated using InfraRed.

IncaRNAtion uses the strategy of multidimensional Boltzmann sampling to target predefined GC content of sampled sequences. By default, InfraRed creates a feature for each function group with value is additive grouped functions values. For illustration purpose, we define a new feature GC to control the proportion of GC content.

```
# By default, Infrared creates a feautre, named 'gc', which sums up
    Functions in the group 'gc'
model.add_feature('GC', 'gc', lambda sample: sum([(c==1 or c==2) for c
    in [sample.values()[i] for i in Xidx]])/seqlen)
```

Finally, we create the sampler using Multidimensional Boltzmann sampling while setting a target with tolerance for feature.

```
# We aim to have 55-65% of GC in each sequence
sampler = ir.MultiDimensionalBoltzmannSampler(model)
sampler.set_target(0.6, 0.05, 'GC')

for i in range(5):
    sample = sampler.targeted_sample()
    print_sample(sample)
```

```
GGGGUUUCUCCACUGGCACAG GC=0.62
UUGUACGACGGGGUGGUCCAC GC=0.62
CGAUCGUGUCGCAUUCCUGGU GC=0.57
GGCGUGAUGUUGUCACGUUGG GC=0.57
UGUUCACGACAGGUGACACGC GC=0.57
```

8.4 FINITE STATE AUTOMATA IN INFRARED

In some applications, it may be needed to impose or to forbid patterns in the sampled sequences. Patterns can be a set of words or a regular expression. One possible approach is adding a rejection step after sampling. However, it could be inefficient when the number of patterns to accept (resp. forbid) is small (resp. large). Another way is to encode as a Constraint Satisfaction Problem (CSP). Since there exists an equivalent Deterministic Finite Automaton (DFA) accpeting the rational language generated by given regular expression [34], we define the design problem as

Problem 8:
Input: DFA $A = (Q, \Sigma, \delta, q_0, Q_F)$, length n
Output: Sequence of length n, $w \in \{A, C, G, U\}^n$, such that w is accepted by A.

Definition 8.10 (Deterministic Finite Automaton): A *Deterministic Finite Automaton* is a 5-tuple $A = (Q, \Sigma, \delta, q_0, Q_F)$ composed of

- A finite set of states Q;
- A finite set of input symbols Σ;
- A transition function $\delta : Q \times \Sigma \to Q$;
- An initial state $q_0 \in Q$;
- A set of final states $Q_F \subseteq Q$.

Let A be a DFA and $w \in \Sigma^n$ be a word of length n. Starting from the initial state, we move from one state to the next, at time t, according to the t-th letter of w and the transition rule. We say w is accepted by A if we are at one of the final states at time $t = n$. More formally, a word $w = w_1 \dots w_n$ is accepted by a DFA if there exists a list

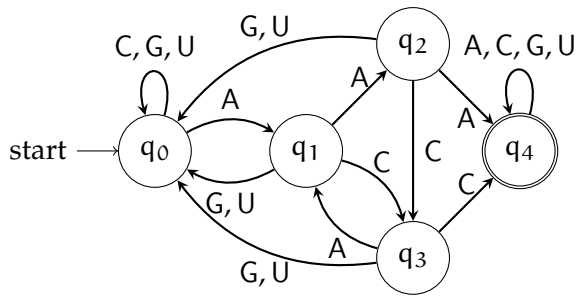


Figure 8.2: Automaton accepting words that contain AAA or ACC.

of states $\mathcal{Y} = \{y_0, \ldots, y_n\} \in \mathcal{Q}^{n+1}$ such that $y_0 = q_0$, $y_n \in \mathcal{Q}_F$, and $\delta(y_{i-1}, w_i) = y_i$ for any $i \in [1, n]$.

The approach is to force InfraRed to sample such a list of states $\mathcal{Y}$ described above for sampled sequence. We need n variables $\mathcal{X} = \{x_1, \ldots, x_n\}$ for sequence, each for one position and $n+1$ variables $\mathcal{Y} = \{y_0, \ldots, y_n\}$ for state list. The domain for variable $x_i \in \mathcal{X}$ is four nucleotides $D(x_i) = \{A, C, G, U\}$. For variable set $\mathcal{Y}$, the first one should be the initial state $D(y_0) = \{q_0\}$ and the last one should be in the final states $D(y_n) = \mathcal{Q}_F$. Other variables can be any state in the automaton, $D(y_i) = \mathcal{Q}$ for $i \in [1, n-1]$. Thus the domain set is,

$$\mathcal{D} = \underbrace{\{A, C, G, U\} \times \cdots \times \{A, C, G, U\}}_{n} \times \{q_0\} \times \underbrace{\mathcal{Q} \times \cdots \times \mathcal{Q}}_{n-1} \times \mathcal{Q}_F.$$

The transition function is turned into the constraint Transition defined as

$$\mathsf{Transition}_{[x_i, y_i, y_{i-1}]}(w_i, q, q') = \begin{cases} \mathsf{True} & \text{if } \delta(q', w_i) = q \\ \mathsf{False} & \text{otherwise.} \end{cases}$$

The constraint Transition is imposed on any two consecutive variables in $\mathcal{Y}$,

$$\mathcal{C} = \{\mathsf{Transition}_{[x_i, y_i, y_{i-1}]};\ i \in [1, n]\}.$$

Problem 8 is then a CSP with the constraint network $(\mathcal{X} \cup \mathcal{Y}, \mathcal{D}, \mathcal{C}, \mathcal{F} = \{\})$. Let w be an assignment returned by InfraRed. The sequence $w_1 \ldots w_n$ is accepted by the automaton A with the list of states $\{w_{n+1}, \ldots, w_{2n+1}\}$.

IMPLEMENTATION As demonstration, we will show how to integrate automaton in IncaRNAtion design problem. Assuming that patterns AAA or ACC should occur at least once in sampled assignment. This is equivalent to generating sequences accepted by the DFA in Figure 8.2 with initial state q_0, final state q_4, and the transition matrix as below.

```
transitions = {
  0: {0: 1, 1: 0, 2: 0, 3: 0},
  1: {0: 2, 1: 3, 2: 0, 3: 0},
  2: {0: 4, 1: 3, 2: 0, 3: 0},
  3: {0: 1, 1: 4, 2: 0, 3: 0},
  4: {0: 4, 1: 4, 2: 4, 3: 4}}
```

We need to introduce new set of $n+1$ variables Y to describe n sequence positions. Each variable has 5 possible values as 5 automaton states.

```
model.add_variables(seqlen+1, 5, 'Y')
```

Furthermore, we need constraints StartState on variable Y_0 and FinalState on variable Y_n to ensure starting with the initial state and ending at the final state. Transition matrix is encoded by the constraint Transition imposed on each state variable, which describes the state transition given a value of proper positional variable. Notice that state variable indices are slightly different than the ones described above since Python is 0-indexed.

```
ir.def_constraint_class('StartState', lambda i: idx([('Y', i)]), lambda
    y: y==0)
ir.def_constraint_class('FinalState', lambda i: idx([('Y', i)]), lambda
    y: y==4)
ir.def_constraint_class('Transition', lambda i: idx([('X',i),('Y',i),('
   Y',i+1)]), lambda x, y1, y2: transitions[y1][x]==y2)

model.add_constraints([StartState(0), FinalState(seqlen)])
model.add_constraints([Transition(i) for i in range(seqlen)])
```

Now, we can sample RNA sequences including AAA or ACC and having a specific GC content.

```
def new_print_sample(sample, n=seqlen):
    values = sample.values()
    seq = ir.rna.values_to_seq(values[:n])
    # model.eval_feature is used to compute feature value instead of
        manual computation since the feature is added in model
    print("{} GC={:.2f} {}".format(seq, model.eval_feature(sample,'GC')
        , ''.join(map(str,values[n:]))))

sampler = ir.MultiDimensionalBoltzmannSampler(model)
sampler.set_target(0.6, 0.05, 'GC' )
for i in range(5):
    sample = sampler.targeted_sample()
    new_print_sample(sample)
```

```
GUGCGCAGUACCGCUACCAGU GC=0.62 0000000100134444444444
GUCCAGUGGACCUCACGGUGA GC=0.62 0000010000134444444444
GCCUCAAAGGUUUGGCCCUUG GC=0.57 0000001244444444444444
UAGUGUCGUUACCGUCGCACG GC=0.57 0010000000013444444444
GCGGGAACUGUACCUAACAGG GC=0.57 0000001230001344444444
```

Each sampled sequence contains at least one AAA or ACC. The last element in each row is the associated sequence of states sampled by InfraRed.

9

RNA POSITIVE AND NEGATIVE DESIGN (RNAPOND)

Despite its flexibility, positive design cannot ensure the absence of alternative stable structures, often preventing the produced sequences to preferentially adopt the target at the thermodynamic equilibrium. By contrast, solutions in the negative design problem must achieve better affinity toward the target than other alternative structures, *i.e.*, possess a small structure defect. In this work, we are interested in the classic inverse folding problem corresponding to negative suboptimal defect ($D^S \leqslant 0$). Inverse folding targets the production of a nucleotide sequence that adopts a targeted structure as its unique MFE structure. Negative and positive design represent distinct tasks, and there is currently no method that efficiently offers the fine level of control enabled by positive design, and the structural specificity of negative design, even though both are typically required in the context of synthetic biology [28, 71].

Our method, called RNAPOND (RNA POsitive and Negative Design), attempts to reconcile positive and negative design, and stems from the following observation: Upon MFE folding, positively-designed RNAs usually differ from their target structure due to the formation of very specific Disruptive Base Pairs (DBPs), that both recurrently represented and reproducible across randomly generated set of design sequences. Examples of such base pairs include helix extensions in both basal and apical regions, or within interior loops, usually associated with a negative selective pressure within RNA multiple sequence alignments [46], and are the object of explicit countermeasures from practitioners of RNA design [36, 51]. This suggests a simple automated strategy that iteratively samples design candidates using positive design principles, identifies a set of dominant DBPs, and forbids them for future iterations unless they induce some inconsistency.

We present in Section 9.1.1 a precise statement of our core computational problems. The problems are further described as Constraint Satisfaction Problem (CSP) for generic sampling framework InfraRed in Section 9.1.2. In Section 9.1.3, we show their integration within RNAPOND for inverse folding problem ($D^S \leqslant 0$). A proof of NP-hardness for the core problems is presented in Section 9.2. In Section 9.3, we perform an empirical assessment of RNAPOND in comparison with the current state of the art.

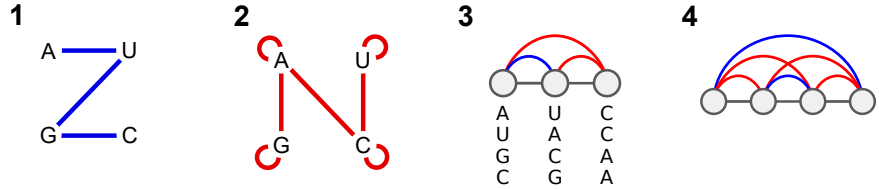


Figure 9.1: Graph representations for set $\mathcal{B}$ of compatible nucleotides pairs (**1**), and set $\overline{\mathcal{B}}$ of incompatible nucleotides (**2**). (**3**) Example of consistent secondary structure $S = \{(1,2)\}$ and disruptive base pairs $\mathcal{R} = \{(1,3),(2,3)\}$, inducing a partition function value of 4 for $\beta = 0$, *i.e.* only 4 out of the 64 possible nucleotide assignments satisfy the constraints induced by S and $\mathcal{R}$. (**4**) Minimal example of inconsistent instance, *i.e.* any RNA sequence violates at least one of the constraints.

9.1 METHOD

9.1.1 *Problem description*

In this work, we identify and forbid Disruptive Base Pairs (DBPs) by forcing their assignment to unpairable nucleotides in $\overline{\mathcal{B}} := \Sigma^2 \setminus \mathcal{B}$ where $\Sigma = \{A, C, G, U\}$ and $\mathcal{B} = \{(C,G),(G,C),(A,U),(U,A),(G,U),(U,G)\}$ (see Figure 9.1). DBPs are base pairs that are recurrent within the stable alternative folds of design candidates generated from positive design principles. Preventing such DBPs from forming is key to satisfying both positive and negative design constraints. A key component of our approach is therefore an algorithm for sampling admissible sequences, defined as compatible with a target input structure S, but also incompatible with a predefined set $\mathcal{R}$ of DBPs. Let us denote by

$$\mathcal{W}_{S,\mathcal{R},n} := \{w \in \Sigma^n \mid S \in \mathcal{S}_w \text{ and } \forall (i,j) \in \mathcal{R}, (w_i, w_j) \in \overline{\mathcal{B}}\}$$

the set of admissible sequences for S and $\mathcal{D}$.

Our approach starts with a preprocessing step, the computation of an (extended dual) partition function over $\mathcal{W}_{S,\mathcal{R},n}$, followed by a stochastic backtrack.

Problem 9 (Extended-Partition-Function):
Input: Secondary structure S, length n, set $\mathcal{R}$ of DBPs, $\beta \in \mathbb{R}^+$
Output: Extended (dual) partition function $\mathcal{Z}(S,\mathcal{R},n) \in \mathbb{R}^+$ such that

$$\mathcal{Z}(S,\mathcal{R},n) = \sum_{w \in \mathcal{W}_{S,\mathcal{R},n}} e^{-\beta . E(w,S)}.$$

The decision version of this problem asks whether there exists a sequence compatible with constraints induced by S and $\mathcal{R}$, *i.e.* whether the admissible sequence set is empty ($|\mathcal{W}_{S,\mathcal{R},n}| > 0 \Rightarrow$ True, $\mathcal{W}_{S,\mathcal{R},n} = \varnothing \Rightarrow$ False). A solution for the problem is used to determine whether a DBP can be added in the set $\mathcal{R}$ (see item 2). We show that the associated decision problem is NP-hard in Section 9.2.

9.1.2 EXTENDED-PARTITION-FUNCTION *as Constraint Satisfaction Problem*

To work around the hardness, we describe the problem as an instance of our CSP framework and use InfraRed to sample design candidates from $\mathcal{W}_{S,\mathcal{D},n}$. Let S be the target structure of length n and $\mathcal{R}$ be the set of DBPs. The variable set is a set of n variables, each for one position and associated with the domain set Σ, $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\mathcal{D} = \Sigma^n$. The constraint set

$$\mathcal{C} = \{\mathsf{Complement}_{[x_i,x_j]};\ (i,j) \in S\} \cup \{\mathsf{DisruptiveBP}_{[x_i,x_j]};\ (i,j) \in \mathcal{R}\}$$

consists of two types of constraint, constraint Complement is imposed on each base pair in the target and constraint DisruptiveBP for each DBP in $\mathcal{R}$ with

$$\mathsf{DisruptiveBP}_{[x_i,x_j]}(w_i, w_j) = \begin{cases} \mathsf{True} & \text{if } (w_i, w_j) \in \overline{\mathcal{B}} \\ \mathsf{False} & \text{otherwise.} \end{cases}$$

A sequence satisfying all constraints in $\mathcal{C}$ is an element of $\mathcal{W}_{S,\mathcal{R},n}$. Figure 9.1 presents examples where the induced dependency graph is consistent (resp. inconsistent), *i.e.* $|\mathcal{W}_{S,\mathcal{R},n}| > 0$ (resp. $\mathcal{W}_{S,\mathcal{R},n} = \varnothing$).

As for function set, we consider a simple base pair energy model, which assumes the energy contribution to the structure is come from base pairs. The model is introduced in RNARedPrint [36] and has been shown to achieve a high correlation ($R = 0.95$) with Turner energy model. Structure energy is captured by adding function $\mathsf{BPEnergy}_{[x_i,x_j]}$ on each base pair,

$$\mathcal{F}_{\mathsf{BP}} = \{(-\beta, \mathsf{BPEnergy}_{[x_i,x_j]});\ (i.j) \in S\}.$$

In addition, we add function $\mathsf{GCControl}_{[x_i]}$ with a negative weight to disfavor C and G in unpaired region,

$$\mathcal{F}_{\mathsf{GC}} = \{(-\beta_{\mathsf{GC}}, \mathsf{GCControl}_{[x_i]});\ \text{unpaired position } i \in S\}.$$

Since GC base pair is favorable in terms of energy, this decreases the chance of forming unwanted base pair between paired and unpaired region of target structure. Thus, the function set is $\mathcal{F} = \mathcal{F}_{\mathsf{BP}} \cup \mathcal{F}_{\mathsf{GC}}$. Running InfraRed on the constraint network $(\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{F})$ gives a FPT approach for Problem 9.

9.1.3 *Approach for RNA inverse folding*

Our method, named RNA Positive and Negative Design (RNAPOND), is inspired by the manual refinement by humans when tackling the task in practice. Its foundation is inspired by the observation that some base pairs and structural motifs, *e.g.* competing helices, more likely than others interfere with the folding of sequences generated from positive design principles. The key idea of our method is to iteratively identify

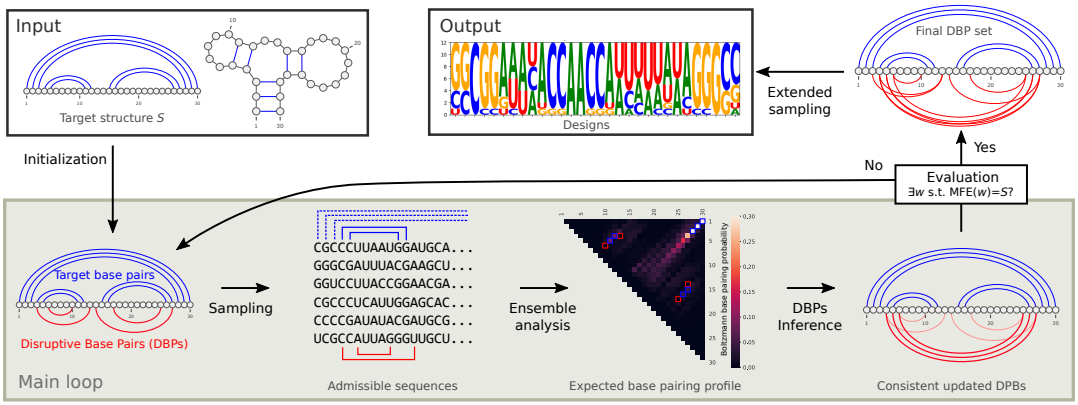


Figure 9.2: General strategy of RNAPOND for the inverse folding of RNA. From an input target structure, an initial set of likely-Disruptive Base Pairs (DBPs) is inferred. At each iteration, new candidate DBPs are inferred by a joint thermodynamic analysis and, if consistent with existing constraints, are added to the list of DBPs. Once some solutions are found, a deeper sampling produces independent and diverse designs for the target.

such recurrent Disruptive Base Pairs (DBPs) and prevent them from occurring in the MFEs of subsequent rounds by adding suitable constraints.

As illustrated in Figure 9.2, RNAPOND takes as input a secondary structure S in dot-bracket notation and, considering a set $\mathcal{R}$ of DBPs that is initialized to helix extensions. We set $\mathcal{R}$ to include base pairs, extending the basal and apical regions of each helix. Then RNAPOND iterates the following steps:

1. **Sampling:** Generate k RNA sequences $\mathbf{w} := w_1, \ldots, w_k$, from the Boltzmann distribution over sequences that are compatible with the secondary structure (S) and DBPs ($\mathcal{R}$). In practice, $k = 200$ sequences per iteration.;

2. **Inference of DBPs:** Identify and add to $\mathcal{R}$ the d (3 in practice) most Disruptive Base Pairs, having highest expected Boltzmann probability within the sample $\mathbf{w}$, such that: a) DBP (i, j) is not in the target structure S; and b) the new constraint network induced by S and $\mathcal{R} \cup \{(i, j)\}$ remains consistent;

3. **Evaluation of candidates:** Compute the MFE structure $S_i^\star$ of each sequence w_i, and report its base-pair distance to S.

These steps are repeated until a solution is found, or the tree width of the dependency graph induced by S and $\mathcal{R}$ exceeds a predefined threshold. Finally, if a solution is found, the method executes a final round of upsampling/evaluation using $K \gg k$, to allow the generation of several (diverse) solutions.

9.2 COMPLEXITY ASPECTS

In the absence of DBPs ($\mathcal{R} = \varnothing$), the compatibility constraints induced by a target structure can always be satisfied [31]. If the target is the open chain ($S = \varnothing$), con-

strained to avoid a set $\mathcal{R}$ of DBPs, then any mononucleotidic sequence will satisfy $\mathcal{R}$, so the problem is again trivially solved by returning True. Combining those two types of constraints constitutes an open problem CONSISTENCY, and turns out to induce substantial computational difficulties.

Problem 10 (CONSISTENCY):
Input: Secondary structure S, length n, set $\mathcal{R}$ of Disruptive Base Pairs
Output: True if there exists a sequence $w \in \{A, C, G, U\}^n$ such that

$$\forall (i,j) \in S,\ (w_i, w_j) \in \mathcal{B} \qquad \text{and} \qquad \forall (i,j) \in \mathcal{R},\ (w_i, w_j) \in \bar{\mathcal{B}},$$

False otherwise.

CONSISTENCY is the decision version of EXTENDED-PARTITION-FUNCTION. The problem is closely related to the linear-time solvable REALIZABILITY of *extended shapes*, considered by Hellmuth, Merkle, and Middendorf [39]. REALIZABILITY considers a set $\mathcal{T}$ of target secondary structures, completed with pseudo base pairs $\mathcal{P}$, and asks if there exists a nucleotide assignment that is compatible with all structures in $\mathcal{T}$, and assigns to one of $\{(A,A),(C,C),(G,G),(U,U)\}$ the content of each pseudo base pair in $\mathcal{P}$. This problem can be solved in linear time by simply testing that the graph, obtained by contracting each pair in $\mathcal{P}$, is bipartite.

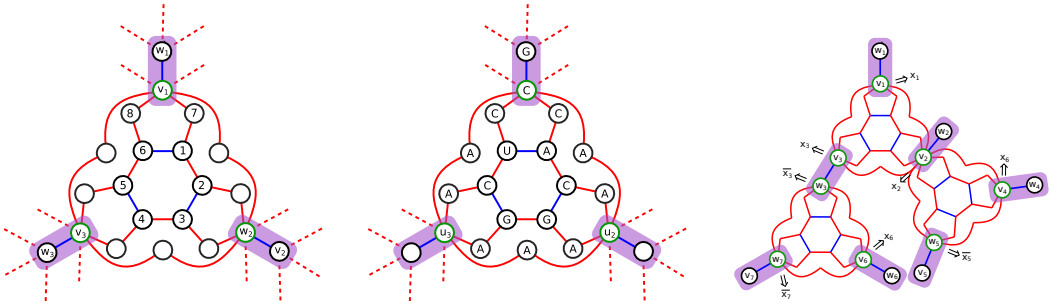
Strikingly, CONSISTENCY only differs from REALIZABILITY, restricted to a single target, in the sense that the positions in DBPs are additionally allowed to take values in $\{(A,G),(G,A),(A,C),(C,A),(U,C),(C,U)\}$. However, this minor difference turns out to be sufficient to greatly increase the computational hardness of the problem.

Theorem 9.1 (NP-hardness of CONSISTENCY): CONSISTENCY *is* NP-*hard.*

Proof. The NP-hardness is shown by a polynomial-time reduction from 3-SAT to CONSISTENCY. Namely, we show that an efficient algorithm for CONSISTENCY would imply that 3-SAT can be solved efficiently, a fact that is highly unlikely as it would imply that $P = NP$.

Problem 11 (3-SAT):
Input: Boolean formula Φ in 3-Conjunctive Normal Form over variables $x_1, \ldots, x_n$
Output: True if a satisfying assignment exists, False otherwise.

Any instance $(S, \mathcal{R})$ of CONSISTENCY can be represented as a dependency graph over positions in $[1, n]$, with edges induced by the target (blue) and disruptive base pairs (red). For a graph or subgraph, an admissible sequence corresponds to an assignment $\mu : V_\Phi \to \Sigma$ of nucleotides to the vertices V_Φ, which satisfies the constraints induced by the edges. For any formula Φ, we construct a graph G_Φ, defined as follows:



(a) (b) (c)

Figure 9.3: (a) Gadget $\mathcal{G}$ for the Boolean clause $(x_1 \vee \bar{x_2} \vee x_3)$. Vertices representing literals of the clause are drawn in green. (b) Design converted from a satisfying assignment of $(l_1 \vee l_2 \vee l_3)$, where l_1 (u_1) is assumed to be True (C). Bases u_2 and u_3 are either C or G based on its value l_2 and l_3 in the assignment. (c) Gadget of formula $(x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee \bar{x_5}) \wedge (\bar{x_3} \vee x_6 \vee \bar{x_7})$.

- For each variable x_i occurring in Φ, G_Φ contains two special variables vertices v_i and w_i, connected with a *blue* base pair (from S);
- Each clause c_k in Φ translates within G_Φ into a gadget $\mathcal{G}_k$, as shown in. Figure 9.3a. $\mathcal{G}_k$ is a subgraph of an instance graph, where 3 vertices (u_1, u_2, u_3) are distinguished to represent the 3 literals in $c_k = (l_1 \vee l_2 \vee l_3)$, each identified with a suitable variable vertex ($u_i = v_i$ if $l_i = x_i$, and $u_i = w_i$ for $l_i = \bar{x}_i$)

Figure 9.3c shows the graph G_Φ obtained for the formula $\Phi = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee \bar{x_5}) \wedge (\bar{x_3} \vee x_6 \vee \bar{x_7})$.

Lemma 9.2: *Let $\mathcal{G}_k$ be one of the gadgets, involving variable vertices (u_1, u_2, u_3). Then an admissible assignment μ exists for $\mathcal{G}_k$ if and only if at least one of $(\mu(u_1), \mu(u_2), \mu(u_3))$ is in $\{A, C\}$.*

Proof. Consider an assignment to $\mathcal{G}_k$. Suppose that the vertices (u_1, u_2, u_3) are all assigned to $\{G, U\}$. Without loss of generality, we assume that u_1 is G (by symmetry of the roles played by G and U).

First, the base pair $(1, 6)$ in the same pentagon as u_1 is either (C, G) or (G, C). Otherwise, one of the bases 1 or 6 must be U; since each nucleotide can pair with G or U, there is no valid value for vertex 8 or 7, and thus no admissible assignment. Next, the outer ring composed by six disruptive base pairs implies that vertices u_2 and u_3 are also G. Therefore, the three base pairs $(1, 2), (3, 4)$, and $(5, 6)$ in the center hexagon are all either (C, G) or (G, C). Without loss of generality, we assume that $(1, 2)$ is (C, G), then $(3, 4)$ and $(5, 6)$ are both (G, C), due to the disruptive base pairs $(2, 3)$ and $(1, 6)$. This yields a contradiction, since the base pair $(4, 5)$ is disruptive, but the nucleotides assigned to 4 and 5 are C and G which can form a base pair. Thus, at least one of the variable vertices must be in $\{C, A\}$.

To conclude, we need to show that assigning one of the variable vertex to $\{A, C\}$ is sufficient to ensure the existence of an admissible assignment. Figure 9.3b shows that assigning A (resp. C) to one of the variable vertices allows the two others to take value G/A/C while maintaining admissibility. □

It can be shown that (u_1, u_2, u_3) can be further restricted to C and G: if there exists an admissible assignment μ for a given triplet $(\mu(u_1), \mu(u_2), \mu(u_3))$, then there also exists μ' that is both admissible, and features the triplet obtained by substituting $A \to C$ and $U \to G$.

Turning to G_Φ, note that any admissible μ induces an admissible assignment for each $\mathcal{G}_k$. It must also feature coherent values on variable nodes, so that $\mu(v_i) \in \{A, C\}$ implies that $\mu(w_i) \in \{G, U\}$ and vice versa. Interpreting $\mu(v_i) \in \{A, C\}$ as $x_i = \text{True}$ (and $\{G, U\}$ as False) we get an assignment that satisfies each clause (Lemma 9.2), and thus satisfies Φ.

Conversely, from a Boolean assignment $(x_1, \ldots, x_n)$ that satisfies Φ, we obtain an assignment μ for G_Φ by setting $(\mu(v_i), \mu(w_i)) := (C, G)$ (resp. (G,C)) if $x_i = \text{False}$ (resp. True). For each gadget $\mathcal{G}_k$, at least one of the variable vertices is True $\to$ C, otherwise c_k would be falsified, and an admissible assignment can be found for the other positions as per Figure 9.3b. Finally, each vertex in G_Φ is connected with at most one blue edge, so there exists an instance $(S_\Phi, \mathcal{R}_\Phi)$ that induces G_Φ (ordering vertices so that blue base pairs in S_Φ do not cross).

We conclude that a solution exists for Φ if and only if there is an admissible sequence for $(S_\Phi, \mathcal{R}_\Phi)$, so that solving CONSISTENCY provides an answer to 3-SAT. Since $(S_\Phi, \mathcal{R}_\Phi)$ has size linear on the number of clauses in Φ, this implies the hardness of CONSISTENCY. □

Moreover, setting $\beta = 0$ leads to $\mathcal{Z}(S, \mathcal{D}, n) = |\mathcal{W}_{S,\mathcal{D},n}|$, so solving EXTENDED-PARTITION-FUNCTION provides an immediate answer to CONSISTENCY.

Corollary 9.3: EXTENDED-PARTITION-FUNCTION *is* NP-*hard.*

Now, given an instance $(S, \mathcal{R}, n)$ we define its dependency graph as:

$$G := ([1, n], S \cup \mathcal{R}).$$

As been seen above, InfraRed provides a solution in time polynomial in $|S|$ and $|\mathcal{R}|$ as long as t, the treewidth of G, is bounded by some constant.

Proposition 9.4: EXTENDED-PARTITION-FUNCTION *is* FPT *for the tree width.*

CONSISTENCY can be solved by setting $\beta = 0$ in the extended problem.

Corollary 9.5: CONSISTENCY *is* FPT *for the tree width of* G.

9.3 VALIDATION AND COMPARISON TO STATE OF THE ART

9.3.1 *Preparation and settings*

DATASETS. We focused our validation effort on two recent collections of target structures:

- The AntaRNA/RFAM dataset [47] consists of a selection of targets extrapolated from RFAM [46] consensuses. For each of the selected family, the consensus structure was mapped onto the smallest sequence of the family, avoiding the artificial insertion of long unpaired regions. We further removed isolated base pairs from those 63 realistic structures, having length range from 36 to 274 nts and showing a typical proportion of paired positions (35% to 80%, median=53%).
- The EteRNA dataset [51] is a collection of 100 artificial *puzzles*, designed to challenge participants of a crowdsourced initiative/gaming. While arguably pathological and not representative of typical design tasks – with certain targets featuring long unpaired regions, very limited structure or even an overwhelming proportion of isolated pairs and stackings – those challenging targets are nevertheless informative as a stress test for our approach, as will be further shown.

BENCHMARK EXECUTION. We considered RNAPOND and selected competitors, including AntaRNA [47], MCTS-RNA [95], MODENA [83], RNAinverse [41], INFO-RNA [12], NUPACK [96], and DSS-OPT [59], invoked with default parameters. Candidate sequences were then validated, computing their MFE structure with the ViennaRNA package 2.4.14, and reporting their base-pair distance to the target.

For the AntaRNA/RFAM dataset, we used the Turner 2004 nearest neighbor model for the tool configuration and verification of candidates. We compared RNAPOND to the state of the art AntaRNA and MCTS-RNA, which both support Turner 2004.

Since the EteRNA game and dataset were designed explicitly for the Turner 1999 model, we used this model for optimization and validation. Some approaches, such as MCTS-RNA and a recent reinforcement learning approach [27], do not currently support the Turner 1999 model, thus we left out of the benchmark. Each software was executed with a time limit of 5 min per instance on a notebook with i7-7500U CPU.

Tools AntaRNA, RNAinverse, INFO-RNA, and NUPACK allow to provide the corresponding parameter file `rna_turner1999.par` from the ViennaRNA package, or provide an explicit option (NUPACK); Others come with compiled-in Turner 1999 parameters (INFO-RNA, DSS-OPT).

In Table 9.1, we show the concrete commands executed for each tool in the benchmark on EteRNA dataset. In particular, we made the following individual choices:

- AntaRNA was executed asking for a single solution for the target structure at the default GC-content.
- RNAinverse was run in a specific mode, where it only returns one solution, when the objective of base pair distance 0 between MFE and target is met perfectly. The target structure is read from the standard input.
- For INFO-RNA, we chose settings analogous to RNAinverse. However, due to an additional termination criterion, it sometimes returns sub-optimal solutions.
- MODENA uses RNAfold in the search path, but does not allow to parametrize it; to utilize Turner 1999 parameters, we run it in combination with ViennaRNA package 1.8.5. Since MODENA returns a set of pareto-optimized solutions, we chose the best one (according to MFE–target distance) as its single solution for the purpose of this benchmark.
- All other tools were run with their default settings.

Tool	Command
AntaRNA	`python2 antaRNA.py -Cstr TARGET -n 1 -P rna_turner1999.par`
RNAinverse	`echo TARGET | RNAinverse -R-1 -P rna_turner1999.par`
INFO-RNA	`echo TARGET | INFO-RNA-2.1.2 -R -1`
MODENA	`echo TARGET >target.in ; modena -f target.in`
DSS-OPT	`opt-md TARGET`
RNAPOND	`RNAPOND.py -n 1 -turner1999 TARGET`
NUPACK	`complexdesign -material rna1999 TARGET`

Table 9.1: Command line calls for the EteRNA dataset benchmark

9.3.2 *Analysis of AntaRNA/RFAM results.*

In terms of success, we observe excellent performances for both RNAPOND, AntaRNA and MCTS-RNA (Figure 9.4). All methods solve all targets, except RF01241, RF00906 and RF00446. For those 3 instances, the best MFE distances to the target are of 1, 1 and 2 respectively for all three tools, suggesting that a solution may simply not exist. Despite the stochastic aspects of RNAPOND, we observed a good level of robustness of our method, with three independent runs showing successful on the exact same instances, and achieving same distance to target otherwise.

A closer look reveals further differences in performance with respect to negative design metrics, including Ensemble Defect [96] (Figure 9.5a) and the Boltzmann probability [60] of the target structure (Figure 9.5b). Interestingly, RNAPOND shows an average normalized ensemble defect of 0.076 and Boltzmann probability of 22.2%, and dominates AntaRNA (0.085/17.8%) with respect to those two metrics, demonstrating its capacity to embrace negative design principles, although MCTS-RNA

(0.056/28.7%) remains superior to both. This trend is inverted when considering the diversity of sequences generated by each tool, as measured by the Positional (Shannon) Entropy, reported in Figure 9.5c. Here RNAPOND (avg 1.6 bits/nt), in its current state, does not match the excellent diversity of AntaRNA (1.95 bits/nt), but greatly exceeds that of MCTS-RNA (1.38 bits/nt).

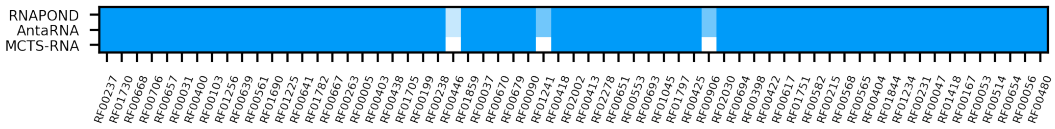


Figure 9.4: Success matrix of RNAPOND and competitors on AntaRNA/RFAM dataset. Dark squares indicate success, *i.e.* at least one solution for the given target/puzzle, and white squares show failure. Lighted shades of blue indicate near success (MFE within 1 or 2 base pairs distance of target). Note that MCTS-RNA does not return any solution in case of failure.

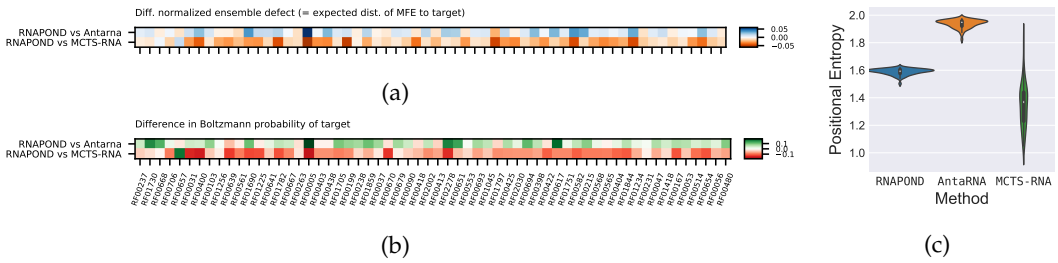


Figure 9.5: Results of RNAPOND and competitors on AntaRNA/RFAM targets. Analysis of (a) ensemble distance, (b) equilibrium probability and (c) sequence diversity of solutions produced for the AntaRNA/RFAM dataset.

9.3.3 *Analysis of EteRNA results.*

On the challenging EteRNA dataset, RNAPOND solves 46 of the 100 instances exactly (MFE structure matching the target). For 5 further instances, we find near-solutions that are close to the target (MFE within 2 BPs of target). For comparison, we report in Figure 9.6 the success, and near-success if available, of several competitors.

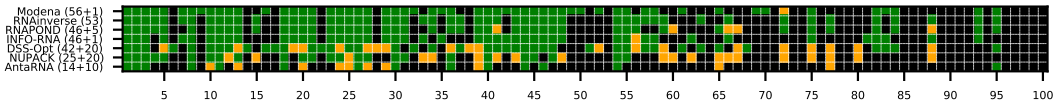
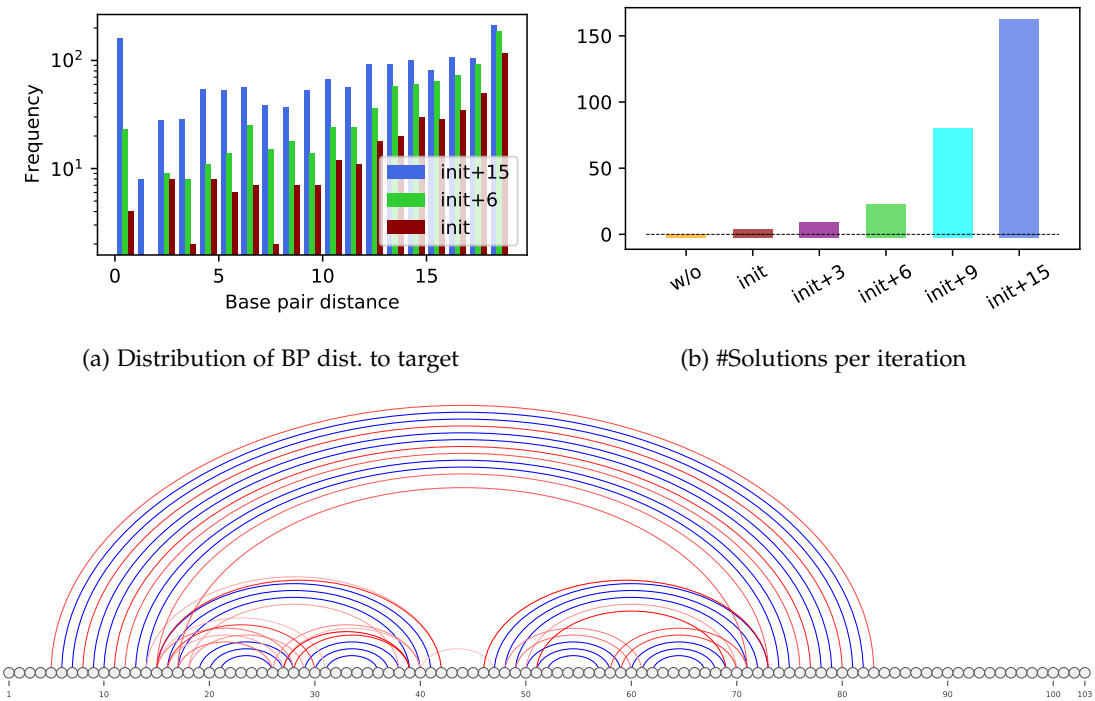


Figure 9.6: Success matrix of RNAPOND and competitors on EteRNA dataset. Green squares indicate success, *i.e.* at least one solution for the given puzzle. Orange squares indicate near success (MFE within 2 BPs of target).

CASE STUDY A – ETERNA 37. We considered this, relatively easy, puzzle to investigate the effect of DBPs on the distribution of distance to the target. Using RNAPOND,

we generated 50 000 samples for the sets of DBPs introduced by the first 5 iterations of RNAPOND (d = 3 DBPs added per round), considering no DBP as a control. As shown in Figure 9.7a, the introduction of DBPs successfully shifts the probability distribution towards solutions, and the probability of sampling a solution appears to increase exponentially (Figure 9.7b), from 0 out of 50 000 in the absence of DBPs to 155 after 5 rounds (init + 15 DBPs). The final set of constraints (DBPs of Figure 9.7c) is sparse, and appears to essentially delimit helices, forbidding their bi-directional extension. Interestingly, this strategy typical of manual design practitioners, and is recovered by RNAPOND despite not being one of its design choice.



(a) Distribution of BP dist. to target

(b) #Solutions per iteration

(c) Final set of disruptive base pairs

Figure 9.7: Illustrating the behavior of RNAPOND on EteRNA puzzle 37 – *"Water Strider"*. (a) Impact of added DBPs on the distribution of distances to target within $50 \cdot 10^3$ sampled sequences; (b) Number of solutions per iteration; (c) Target structure and final set of DBPs.

CASE STUDY B – ETERNA 58. On this example, RNAPOND finds a first solution after generating 9 initial DBPs, supplemented by 30 more DBPs introduced over 10 rounds (Figure 9.8). Remarkably, DBPs are not only introduced to avoid helix extensions, but also unwanted interactions within the large multi-loop. This is achieved through the introduction of key local stack-like DBPs which appear sufficient to break the symmetries presented by the multiloop.

CASE STUDY C – ETERNA 22. This challenging target (see Figure 9.9) consists of 400 unpaired nucleotides, and could not be solved within the time limit. It is easy

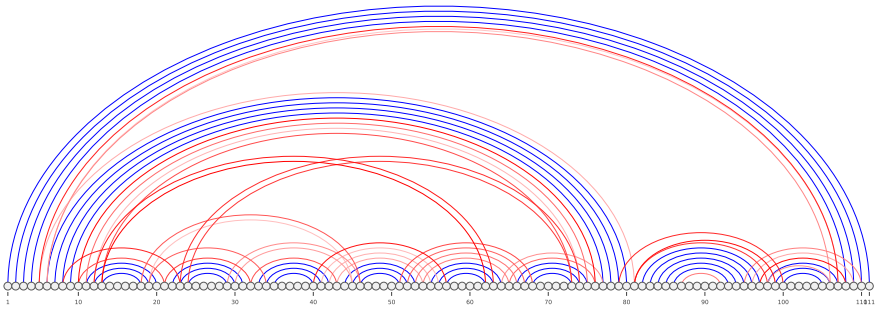
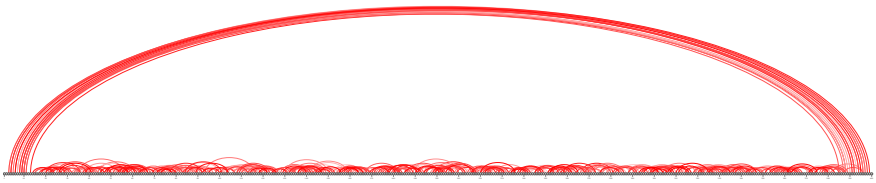
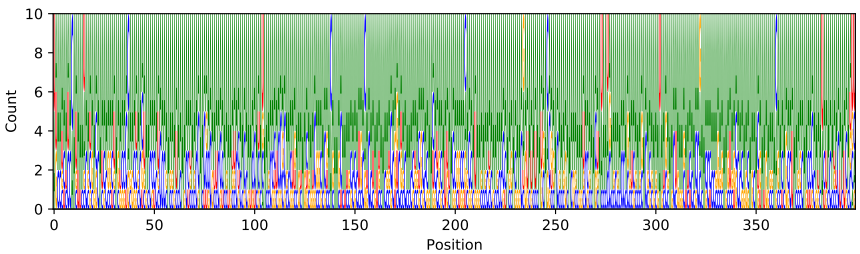


Figure 9.8: EteRNA puzzle 58 – *"Multiloop..."*. Target structure and final set of DBPs after 10 rounds.

to solve manually by only using unpairable nucleotides, but we decided to ignore such an *ad hoc* rule, both to ensure maximal sequence diversity and to preserve the level of generality of RNAPOND. Interestingly, lifting the time limit, yields a solution after four hours and 154 rounds, introducing 462 DBPs (see Figure 9.9a). Those incompatibility constraints are surprisingly local, with the except of DBPs connecting the 5′ and 3′ ends, and suggests that forbidding hairpins of moderate span may be sufficient to design large loops within challenging RNAs. Figure 9.9b shows the sequence logo for ten solutions sampled with the set of 462 DBPs after 154 rounds. Although adenine (in green) highly presents in solution sequences, a good sequence diversity is observed. Indeed, about one forth positions of each solution are different from adenine. It shows the abaility of RNAPOND to generate diverse sequence even for such challenging instance.



(a)



(b)

Figure 9.9: EteRNA puzzle 22 – *"This is ACTUALLY Small And Easy"*. (a) Target structure and final set of DBPs after 154 rounds. (b) Sequence logo for 10 designs returned by RNAPOND. The logo is generated using Logomaker [84] with green for A, blue for C, yellow for G, and red for U.

Part IV

CONCLUSION AND PERSPECTIVES

10 CONCLUSION AND PERSPECTIVES

10.1 CONCLUSION

In this thesis, we tackled the RNA design problem from two different angles. We examined the impact of local structural motifs on the negative designability of secondary structures for the first part of the work. Since a structure is locally undesignable, globally, the structure is also undesignable. We attempted to estimate an upper bound for designable structures given local obstructions. For the second part, we studied an application of tree decomposition in the RNA design problem. We attempted to generalize a framework for positive multi-targets design and studied an application for the negative design.

We began by describing a procedure for computing a set of local obstructions. The presence of such a local motif implies obstruction of designability for a secondary structure. It holds for any negative design objective expressed as a monotonic defect over loops. We obtained, using the Turner energy model, a local obstruction database with a length up to 14 for different defects and tolerances, representing different secondary structure prediction paradigms. The exponential growth of the local obstruction set with the length of investigated motifs indicates the exponential nature may be intrinsic to the problem. We also showed preliminary results on searching hard or undesignable motifs in experimentally determined structures. It suggests a possible negative selective pressure on these motifs.

As a first approach for estimating the number of designable structures, we enumerated secondary structures, avoiding a local obstruction set up an upper bound. We constructed a grammar integrated with the subtraction of structures having a root occurrence of local obstruction. With the classic analytic combinatorics techniques, we computed the asymptotic upper bound in an automated process. It reveals an overall sparsity within the entire structure space for the designable structures. The number of designable structures increases exponentially with the structure length but much slower than initially anticipated.

Next, we were interested in estimating the structure ensemble defect distribution. We have obtained a closed-form expression for a minimum distance of 0, given an overlap-free motif set. In a more realistic case with a minimum distance of 3, we have proposed an estimation from the empirical distribution given the complete motif set. In both cases, we have shown that the ensemble defect follows a Gaussian limiting distribution with the mean linear to structure length n. Therefore, with a commonly used ensemble defect tolerance $0.01n$, the proportion of designable secondary structure is even less than the upper bound obtained using the first approach with a tolerance of 1.

For the second part of this thesis, we attempted to find RNA sequences for both positive and negative design objectives. We have developed InfraRed, a generic framework of Fixed-Parameter Trackable Boltzmann-weighted sampling. The framework is designed for the extended Constraint Satisfaction Problem, a generalization of the design problem, to enable flexible design goals. We have provided an interface written in Python to give users easy access to define their design problems. One can realize the positive design by introducing energy functions into the problem. We have also shown that the sampling is not limited to sequence with an example of automaton integration.

Finally, we have presented an application of Infrared. As implemented in RNAPOND, we have proposed a global sampling approach to reconcile positive and negative design. The approach achieves the negative design without altering the Boltzmann-weighted distribution by preventing two assigned nucleotides from forming a base pair in a disruptive base pair. Even though the essential problem of finding compatible sequence is NP-hard, RNAPOND obtains good designs with an efficient iterative sequence sampling using InfraRed.

Testing on two different benchmarks, we have shown that our method, RNAPOND, has a comparable performance with state-of-the-art. Overall, these results support the notion that RNAPOND achieves an excellent trade-off between exploration, the optimization of negative design criteria, and exploitation, witnessed by a sizable sequence diversity. The success of RNAPOND in designing complex RNA architectures suggests that a local selective pressure may often be sufficient to implement negative design principles, allowing the evolution of complex RNA architectures.

10.2 PERSPECTIVES

10.2.1 *Extensions of local obstruction study*

Since NUPACK [96] considers the ensemble defect as the objective function to minimize for local search, an upper bound of ensemble defect can be computed as shown in Figure 10.1. The difference between the lower and upper bound distribution suggests further improvement in estimation. One of the possible directions is to take motif overlap into account while constructing the grammar. In addition, it requires a solution to calculate the ensemble defect for overlapped motifs.

Another potential extension is about computational approach for design problem. Algorithm 5.2, which enumerates local obstruction, can easily be modified to keep the suitable candidate design sequences for each designable motif. These precomputed candidate sequences greatly restrict classic design algorithms' search space and suggest a strategy for hard design instances. As an illustration, while investigating our database of local obstructions, we discovered that lonely base pairs appear in a few designable motifs, usually considered unstable in the Turner model and challenging to design. For example, the structure `(((.(....).)))` is the MFE

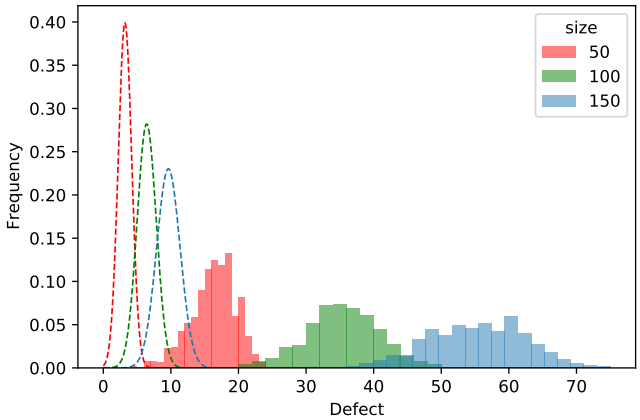


Figure 10.1: Empirical distribution of ensemble defect computed by NUPACK. For each length, the distribution is computed over 10 000 secondary structures with $\theta = 3$ uniformly genarated by GenRGenS. Dashed curves are defect lower bound $\tilde{D}$ estimated in Section 7.3.2 ($\mu = 0.065$ and $\sigma^2 = 0.02$).

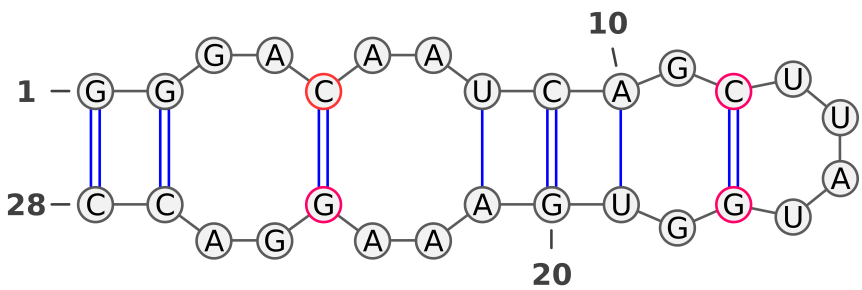


Figure 10.2: Example of secondary structure with isolated base pairs (red) in its MFE structure, obtained by connecting local solutions.

structure of the RNA sequence UCAGCUUAUGGUGA. We also found that the motif `((..(*)..))` could be designable for some collection of sequences. Combining sequences adopting these two motifs, as presented in Figure 10.2, we could verify that the RNA sequence

GGGACAAUCAGCUUAUGGUGAAAGGACC

is predicted by RNAfold to adopt its unique MFE structure of

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

featuring two isolated base pairs, and a free-energy of -6.4kcal.mol^{-1}, a stability unmatched across several runs of RNAinverse [41] and NUPACK [96]. While this observation remains anecdotal, it is supported by the success of recent approaches using (partial) libraries of local motifs [4].

10.2.2 *Towards pseudoknotted and tertiary structure*

Our study on enumerating designable secondary structures is easy to apply to pseudoknotted structures. Several grammars exist to describe different algebraic types of pseudoknots and to enable the characterization with generating functions [64]. To design a pseudoknot using Infrared, one can add new constraints for pseudoknotted base pairs into the framework.

While folding RNA tertiary structure, non-canonical base pairs are formed and stabilize loops. These non-canonical base pairs with associated secondary loops, called RNA 3D modules, have thus been considered as significant components for RNA structure. Unlike the secondary level, currently, we lack a well-described energy model. An alternative way is needed to integrate RNA module messages into the design framework.

One of the means to recognize modules in a given sequence is treating a module as a Bayesian network [17, 72]. Each node, also called a variable, represents a base of a module. The probability of observing a nucleotide at a base is computed with a conditional dependencies probability estimated from sequence data. A naive way is to add a function for each base in the InfraRed framework to encode the associated conditional dependencies probability table. However, the number of dependent nucleotides in a module is usually large, implying large treewidth. In the recent work for module identification [73], in collaboration with R. Sarrazin-Gendron, we decreased greatly the module dependency using tree decomposition. Such reduction suggests a potential application of InfraRed to sample RNA sequences towards the tertiary structure.

10.2.3 *Extension of InfraRed*

Our framework, InfraRed, follows the same policy for secondary structure sampling, as introduced in Section 2.3, but uses a more complicated decomposition, *i.e.*, tree decomposition. Indeed, both rely on dynamic programming to calculate the (dual) partition function and stochastic backtrack to sample sequences/structures. Therefore, one of the further developments is implementing RNA secondary structure-related approaches presented in Chapter 2 in the framework for sequence, such as finding sequence minimizing target free-energy, returning all suboptimal sequences within a range from the MFE, and non-redundant sequence sampling.

In Section 8.4, we presented the integration of automata in the InfraRed framework. This allows us to consider more flexible design problems related to sequence patterns. For example, we can use new variables to record the pattern (overlapped) occurrences in the sampled sequence with a slight modification. One of the applications is then designing sequences containing a low number of a given k-mer.

10.2.4 *Extension of RNAPOND*

One of the possible extensions of RNAPOND is to explore the assignment neighborhood for further refinement. Combining with local search heuristics makes the method a *glocal* approach [69]. The critical point is to maintain the compatibility of mutated assignments. Indeed, randomly changing a nucleotide may produce a chain of nucleotide modifications concerning several (disruptive) base pairs. It can also lead to inconsistency with target structure and disruptive base pairs.

The choice of disruptive base pairs is sometimes decisive. Other *ad hoc* initialization strategies may improve the performance, such as restricting undesired interactions within a multi-loop, as seen in Figure 9.8. Furthermore, a poor choice of disruptive base pairs during the iteration can end up with large treewidth. A simple solution is to restart the iteration. Recently, an approach for treewidth reduction has been published [57]. Authors showed, as a proof-of-concept, a decrease of treewidth from 9 to 7 by removing only 3 out of 183 Disruptive Base Pairs (DBPs) produced by RNAPOND for the puzzle EteRNA 77. This preliminary result suggests a strategy to overcome the hardness by partially restarting the iteration with a reduced DBP list.

BIBLIOGRAPHY

[1] Rosalía Aguirre-Hernández, Holger H Hoos, and Anne Condon. "Computational RNA secondary structure design: empirical complexity and improved methods." In: *BMC Bioinformatics* 8 (2007), p. 34. DOI: 10.1186/1471-2105-8-34.

[2] Mirela Andronescu, Anthony P. Fejes, Frank Hutter, Holger H. Hoos, and Anne Condon. "A new algorithm for RNA secondary structure design." In: *Journal of Molecular Biology* 336.3 (2004), pp. 607–624. DOI: 10.1016/j.jmb.2003.12.041.

[3] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. "Complexity of Finding Embeddings in a k-Tree." In: *SIAM Journal on Algebraic Discrete Methods* 8.2 (1987), pp. 277–284. DOI: 10.1137/0608024.

[4] Stanislav Bellaousov, Mohammad Kayedkhordeh, Raymond J Peterson, and David H Mathews. "Accelerated RNA secondary structure design using preselected sequences for helices and loops." In: *RNA (New York, N.Y.)* 24 (11 Nov. 2018), pp. 1555–1567. ISSN: 1469-9001. DOI: 10.1261/rna.066324.118.

[5] Olivier Bodini and Yann Ponty. "Multi-dimensional Boltzmann Sampling of Languages." In: *21st International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms (AofA'10)*. Ed. by Michael Drmota and Bernhard Gittenberger. Vol. DMTCS Proceedings vol. AM, 21st International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms (AofA'10). DMTCS Proceedings. Vienna, Austria: Discrete Mathematics and Theoretical Computer Science, June 2010, pp. 49–64. URL: https://hal.inria.fr/hal-00450763.

[6] Hans L. Bodlaender and Arie M.C.A. Koster. "Treewidth computations I. Upper bounds." In: *Information and Computation* 208.3 (2010), pp. 259–275. DOI: 10.1016/j.ic.2009.03.008.

[7] Édouard Bonnet, Paweł Rzążewski, and Florian Sikora. "Designing RNA Secondary Structures Is Hard." In: *Research in Computational Molecular Biology - 22nd Annual International Conference, RECOMB 2018*. Ed. by Benjamin J. Raphael. Vol. 10812. Lecture Notes in Computer Science. Paris: Springer, 2018, pp. 248–250.

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

[9] Ravi Boppana and Magnús M Halldórsson. "Approximating maximum independent sets by excluding subgraphs." In: *BIT Numerical Mathematics* 32.2 (1992), pp. 180–196.

[10] M Bousquet-Melou. "Convex polyominoes and algebraic languages." In: *Journal of Physics A: Mathematical and General* 25.7 (1992), pp. 1935–1944. DOI: 10.1088/0305-4470/25/7/032. URL: https://doi.org/10.1088/0305-4470/25/7/032.

[11] Stephen K Burley *et al.* "RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences." In: *Nucleic Acids Research* 49.D1 (2020), pp. D437–D451. DOI: 10.1093/nar/gkaa1038.

[12] Anke Busch and Rolf Backofen. "INFO-RNA—a fast approach to inverse RNA folding." In: *Bioinformatics* 22.15 (2006), pp. 1823–31. DOI: 10.1093/bioinformatics/btl194.

[13] Frédéric Chyzak, Michael Drmota, Thomas Klausner, and Gerard Kok. "The distribution of patterns in random trees." In: *Combinatorics, Probability and Computing* 17.1 (2008), pp. 21–59.

[14] Gwendal Colleta, Julien Davidb, and Alice Jacquotb. "Random Sampling of Ordered Trees according to the Number of Occurrences of a Pattern." In: (Nov. 2014).

[15] Rémi Coulom. "Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search." In: *5th International Conference on Computer and Games*. Ed. by Paolo Ciancarini and H. Jaap van den Herik. Turin, Italy, May 2006. URL: https://hal.inria.fr/inria-00116992.

[16] F.H.C. Crick. "Codon—anticodon pairing: The wobble hypothesis." In: *Journal of Molecular Biology* 19.2 (1966), pp. 548–555. DOI: 10.1016/s0022-2836(66)80022-0.

[17] José Almeida Cruz and Eric Westhof. "Sequence-based identification of 3D structural modules in RNA with RMDetect." In: *Nat Methods* 8.6 (2011), pp. 513–21. DOI: 10.1038/nmeth.1603.

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

[19] Rina Dechter. "Tractable Structures for Constraint Satisfaction Problems." In: *Handbook of Constraint Programming*. Elsevier, 2006, pp. 209–244. DOI: 10.1016/s1574-6526(06)80011-8.

[20] Alain Denise, Yann Ponty, and Michel Termier. "Controlled non-uniform random generation of decomposable structures." In: *Theoretical Computer Science* 411.40 (2010), pp. 3527–3552.

[21] Thomas van Dijk, Jan-Pieter van den Heuvel, and Wouter Slob. "Computing treewidth with LibTW." In: *Citeseer. http://citeseerx. ist. psu. edu/viewdoc/download* (2006).

[22] Ye Ding and Charles E Lawrence. "A statistical sampling algorithm for RNA secondary structure prediction." In: *Nucleic acids research* 31 (24 Dec. 2003), pp. 7280–7301. ISSN: 1362-4962. DOI: 10.1093/nar/gkg938.

[23] Gesine Domin, Sven Findeiß, Manja Wachsmuth, Sebastian Will, Peter F. Stadler, and Mario Mörl. "Applicability of a computational design approach for synthetic riboswitches." In: (2016), gkw1267. DOI: 10.1093/nar/gkw1267.

[24] Ivan Dotu, William A. Lorenz, Pascal Van Hentenryck, and Peter Clote. "Computing folding pathways between RNA secondary structures." In: *Nucleic Acids Research* 38.5 (2009), pp. 1711–1722. DOI: 10.1093/nar/gkp1054.

[25] Michael Drmota. "Systems of functional equations." In: *Random Structures & Algorithms* 10.1-2 (1997), pp. 103–124.

[26] Jérémie Du Boisberranger, Danièle Gardy, and Yann Ponty. "The weighted words collector." In: *International Meeting on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (AOFA 2012)*. Ed. by France) Nicolas Broutin (INRIA and Canada) Luc Devroye (McGill. Vol. AQ. Discrete Mathematics & Theoretical Computer Science. Episciences.org, 2012, pp. 243–264.

[27] Peter Eastman, Jade Shi, Bharath Ramsundar, and Vijay S. Pande. "Solving the RNA design problem with reinforcement learning." In: *PLOS Computational Biology* 14.6 (June 2018), pp. 1–15. DOI: 10.1371/journal.pcbi.1006176.

[28] Sven Findeiß, Maja Etzel, Sebastian Will, Mario Mörl, and Peter F Stadler. "Design of Artificial Riboswitches as Biosensors." In: *Sensors (Basel, Switzerland)* 17.9 (9 Aug. 2017), E1990. ISSN: 1424-8220. DOI: 10.3390/s17091990.

[29] Philippe Flajolet and Andrew M. Odlyzko. "Singularity Analysis of Generating Functions." In: *SIAM J. Discrete Math.* 3.2 (1990), pp. 216–240. DOI: 10.1137/0403019. URL: https://doi.org/10.1137/0403019.

[30] Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics*. cambridge University press, 2009.

[31] C. Flamm, I. L Hofacker, S. Maurer-Stroh, P. F Stadler, and M. Zehl. "Design of multistable RNA molecules." In: *RNA (New York, N.Y.)* 7 (2 2001), pp. 254–265. ISSN: 1355-8382.

[32] Juan Antonio Garcia-Martin, Peter Clote, and Ivan Dotu. "RNAiFOLD: a constraint programming algorithm for RNA inverse folding and molecular design." In: *Journal of Bioinformatics and Computational Biology* 11.2 (2013), p. 1350001. DOI: 10.1142/S0219720013500017.

[33] M. R. Garey and D. S. Johnson. "`` Strong '' NP-Completeness Results." In: *Journal of the ACM* 25.3 (1978), pp. 499–508. DOI: 10.1145/322077.322090.

[34] Jozef Haleš, Alice Héliou, Ján Maňuch, Yann Ponty, and Ladislav Stacho. "Combinatorial RNA Design: Designability and Structure-Approximating Algorithm in Watson-Crick and Nussinov-Jacobson Energy Models." In: *Algorithmica* 79.3 (2017), pp. 835–856. DOI: 10.1007/s00453-016-0196-x.

[35] Stefan Hammer, Birgit Tschiatschek, Christoph Flamm, Ivo L Hofacker, and Sven Findeiß. "RNAblueprint: flexible multiple target nucleic acid sequence design." In: *Bioinformatics* 33.18 (2017). Ed. by Cenk Sahinalp, pp. 2850–2858. DOI: 10.1093/bioinformatics/btx263.

[36] Stefan Hammer, Christian Günzel, Mario Mörl, and Sven Findeiß. "Evolving methods for rational de novo design of functional RNA molecules." In: *Methods* 161 (2019). Development and engineering of artificial RNAs, pp. 54 –63. ISSN: 1046-2023. DOI: https://doi.org/10.1016/j.ymeth.2019.04.022. URL: http://www.sciencedirect.com/science/article/pii/S1046202318302895.

[37] Stefan Hammer, Wei Wang, Sebastian Will, and Yann Ponty. "Fixed-parameter tractable sampling for RNA design with multiple target structures." In: *BMC Bioinformatics* 20.1 (Dec. 2019), p. 209. DOI: 10.1186/s12859-019-2784-7. URL: https://hal.inria.fr/hal-02112888.

[38] Christine E Heitsch. "Combinatorics on plane trees, motivated by RNA secondary structure configurations." In: *preprint* (2006).

[39] Marc Hellmuth, Daniel Merkle, and Martin Middendorf. "Extended shapes for the combinatorial design of RNA sequences." In: *Int J Comput Biol Drug Des* 2.4 (2009), pp. 371–84. ISSN: 1756-0756. DOI: 10.1504/IJCBDD.2009.030767.

[40] Paul G. Higgs and Niles Lehman. "The RNA World: molecular cooperation at the origins of life." In: *Nature Reviews Genetics* 16.1 (2014), pp. 7–17. DOI: 10.1038/nrg3841.

[41] I. L. Hofacker, W. Fontana, P. Stadler, L. Bonhoeffer, M. Tacker, and P. Schuster. "Fast folding and comparison of RNA secondary structures." In: *Monatshefte für Chemie / Chemical Monthly* 125.2 (1994), pp. 167–188. DOI: 10.1007/BF00818163.

[42] Ivo L Hofacker, Peter Schuster, and Peter F Stadler. "Combinatorics of RNA secondary structures." In: *Discrete Applied Mathematics* 88.1-3 (1998), pp. 207–237.

[43] Heike Hofmann, Karen Kafadar, and Hadley Wickham. *Letter-value plots: Boxplots for large data*. Tech. rep. had.co.nz, 2011.

[44] Xue Huang, Wei Sun, Zhi Cheng, Minxuan Chen, Xueyan Li, Jiuyu Wang, Gang Sheng, Weimin Gong, and Yanli Wang. "Structural basis for two metal-ion catalysis of DNA cleavage by Cas12i2." In: *Nature Communications* 11.1 (2020). DOI: 10.1038/s41467-020-19072-6.

[45] Thomas Jörg, Olivier C Martin, and Andreas Wagner. "Neutral network sizes of biological RNA molecules can be computed and are not atypically small." In: *BMC bioinformatics* 9.1 (2008), pp. 1–12.

[46] Ioanna Kalvari, Joanna Argasinska, Natalia Quinones-Olvera, Eric P Nawrocki, Elena Rivas, Sean R Eddy, Alex Bateman, Robert D Finn, and Anton I Petrov. "Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families." In: *Nucleic acids research* 46 (D1 Jan. 2018), pp. D335–D342. ISSN: 1362-4962. DOI: 10.1093/nar/gkx1038.

[47] Robert Kleinkauf, Martin Mann, and Rolf Backofen. "antaRNA: ant colony-based RNA sequence design." In: *Bioinformatics (Oxford, England)* 31 (19 Oct. 2015), pp. 3114–3121. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btv319.

[48] B. Knudsen and J. Hein. "RNA secondary structure prediction using stochastic context-free grammars and evolutionary history." In: *Bioinformatics* 15.6 (1999), pp. 446–454. DOI: 10.1093/bioinformatics/15.6.446.

[49] Tadeusz Kulinski, Mikolaj Olejniczak, Hendrik Huthoff, Lukasz Bielecki, Katarzyna Pachulska-Wieczorek, Atze T. Das, Ben Berkhout, and Ryszard W. Adamiak. "The Apical Loop of the HIV-1 TAR RNA Hairpin Is Stabilized by a Cross-loop Base Pair." In: *Journal of Biological Chemistry* 278.40 (2003), pp. 38892–38901. DOI: 10.1074/jbc.m301939200.

[50] Steven P. Lalley. "Finite Range Random Walk on Free Groups and Homogeneous Trees." In: *The Annals of Probability* 21.4 (1993). DOI: 10.1214/aop/1176989012.

[51] Jeehyung Lee *et al.* "RNA design rules from a massive open laboratory." In: *Proceedings of the National Academy of Sciences U S A* 111.6 (2014), pp. 2122–2127. DOI: 10.1073/pnas.1313039111.

[52] Neocles B Leontis and Eric Westhof. "Geometric nomenclature and classification of RNA base pairs." In: *Rna* 7.4 (2001), pp. 499–512.

[53] William Andrew Lorenz, Peter Clote, and Yann Ponty. "Asymptotics of RNA shapes." In: *Journal of Computational Biology* 15.1 (2008), pp. 31–63. DOI: 10.1089/cmb.2006.0153. URL: https://hal.inria.fr/inria-00548861.

[54] Xiang-Jun Lu, Harmen J. Bussemaker, and Wilma K. Olson. "DSSR: an integrated software tool for dissecting the spatial structure of RNA." In: *Nucleic Acids Research* (2015), gkv716. DOI: 10.1093/nar/gkv716.

[55] Zhi John Lu, Jason W Gloor, and David H Mathews. "Improved RNA secondary structure prediction by maximizing expected pair accuracy." In: *RNA (New York, N.Y.)* 15 (10 Oct. 2009), pp. 1805–1813. ISSN: 1469-9001. DOI: 10.1261/rna.1643609.

[56] Ján Maňuch, Chris Thachuk, Ladislav Stacho, and Anne Condon. "NP-completeness of the energy barrier problem without pseudoknots and temporary arcs." In: *Natural Computing* 10.1 (2010), pp. 391–405. DOI: 10.1007/s11047-010-9239-4.

[57] Bertrand Marchand, Yann Ponty, and Laurent Bulteau. "Tree Diet: Reducing the Treewidth to Unlock FPT Algorithms in RNA Bioinformatics." In: *WABI 2021 - 21st Workshop on Algorithms in Bioinformatics*. Paris, France, 2021. URL: https://hal.inria.fr/hal-03206132.

[58] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner. "Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure." In: *Journal of Molecular Biology* 288.5 (1999), pp. 911–940. DOI: 10.1006/jmbi.1999.2700.

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

[60] J. S. McCaskill. "The equilibrium partition function and base pair binding probabilities for RNA secondary structure." In: *Biopolymers* 29.6-7 (1990), pp. 1105–1119. DOI: 10.1002/bip.360290621.

[61] A Meir and J.W Moon. "On an asymptotic method in enumeration." In: *Journal of Combinatorial Theory, Series A* 51.1 (1989), pp. 77 –89. ISSN: 0097-3165. DOI: https://doi.org/10.1016/0097-3165(89)90078-2. URL: http://www.sciencedirect.com/science/article/pii/0097316589900782.

[62] Zhichao Miao *et al.* "RNA-Puzzles Round III: 3D RNA structure prediction of five riboswitches and one ribozyme." In: *RNA* 23.5 (Apr. 2017), pp. 655–672. DOI: 10.1261/rna.060368.116. URL: https://hal.archives-ouvertes.fr/hal-02171291.

[63] Juraj Michálik, Hélène Touzet, and Yann Ponty. "Efficient approximations of RNA kinetics landscape using non-redundant sampling." In: *Bioinformatics (Oxford, England)* 33 (14 July 2017), pp. i283–i292. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btx269.

[64] Markus E Nebel and Frank Weinberg. "Algebraic and combinatorial properties of common RNA pseudoknot classes with applications." In: *Journal of computational biology : a journal of computational molecular cell biology* 19 (10 Oct. 2012), pp. 1134–1150. ISSN: 1557-8666. DOI: 10.1089/cmb.2011.0094.

[65] R. Nussinov and A.B. Jacobson. "Fast algorithm for predicting the secondary structure of single-stranded RNA." In: *Proceedings of the National Academy of Sciences U S A* 77 (1980), pp. 6903–13.

[66] Yann Ponty. "Efficient sampling of RNA secondary structures from the Boltzmann ensemble of low-energy." In: *Journal of Mathematical Biology* 56.1-2 (2007), pp. 107–127. DOI: 10.1007/s00285-007-0137-z.

[67] Svetlana Poznanović and Christine E. Heitsch. "Asymptotic distribution of motifs in a stochastic context-free grammar model of RNA folding." In: *Journal of Mathematical Biology* 69.6-7 (2014), pp. 1743–1772. DOI: 10.1007/s00285-013-0750-y.

[68] Lei S. Qi and Adam P. Arkin. "A versatile framework for microbial engineering using synthetic non-coding RNAs." In: *Nature Reviews Microbiology* 12.5 (2014), pp. 341–354. DOI: 10.1038/nrmicro3244.

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

[70] Vladimir Reinharz, Antoine Soulé, Eric Westhof, Jérôme Waldispühl, and Alain Denise. "Mining for recurrent long-range interactions in RNA structures reveals embedded hierarchies in network families." In: *Nucleic Acids Research* 46.8 (2018), pp. 3841–3851.

[71] Guillermo Rodrigo, Thomas E. Landrain, Eszter Majer, José-Antonio Daròs, and Alfonso Jaramillo. "Full Design Automation of Multi-State RNA Devices to Program Gene Expression Using Energy-Based Optimization." In: *PLoS Computational Biology* 9.8 (2013), e1003172. DOI: 10.1371/journal.pcbi.1003172.

[72] Roman Sarrazin-Gendron, Vladimir Reinharz, Carlos G Oliver, Nicolas Moitessier, and Jérôme Waldispühl. "Automated, customizable and efficient identification of 3D base pair modules with BayesPairing." In: *Nucleic acids research* (2019).

[73] Roman Sarrazin-Gendron, Hua-Ting Yao, Vladimir Reinharz, Carlos G Oliver, Yann Ponty, and Jérôme Waldispühl. "Stochastic Sampling of Structural Contexts Improves the Scalability and Accuracy of RNA 3D Modules Identification." In: *RECOMB 2020 - 24th Annual International Conference on Research in Computational Molecular Biology*. Proceedings of RECOMB - 24th Annual International Conference on Research in Computational Molecular Biology - 2020. Padova, Italy, May 2020. URL: https://hal.inria.fr/hal-02354733.

[74] Thomas Schiex, Helene Fargier, and Gerard Verfaillie. "Valued Constraint Satisfaction Problems: Hard and Easy Problems." In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*. IJCAI'95. Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc., 1995, 631–637. ISBN: 1558603638.

[75] LLC Schrödinger. *The PyMOL molecular graphics system, version 1.8*. 2015.

[76] Claudia Steglich, Debbie Lindell, Matthias Futschik, Trent Rector, Robert Steen, and Sallie W Chisholm. "Short RNA half-lives in the slow-growing marine cyanobacterium Prochlorococcus." In: *Genome Biology* 11.5 (2010), R54. DOI: 10.1186/gb-2010-11-5-r54.

[77] P.R. Stein and M.S. Waterman. "On some new sequences generalizing the Catalan and Motzkin numbers." In: *Discrete Mathematics* 26.3 (1979), pp. 261–272. ISSN: 0012-365X. DOI: https://doi.org/10.1016/0012-365X(79)90033-5. URL: http://www.sciencedirect.com/science/article/pii/0012365X79900335.

[78] Jean-Marc Steyaert and Philippe Flajolet. "Patterns and pattern-matching in trees: an analysis." In: *Information and Control* 58.1-3 (1983), pp. 19–58.

[79] Defne Surujon, Yann Ponty, and Peter Clote. "Small-World Networks and RNA Secondary Structures." In: *Journal of Computational Biology* 26.1 (2019), pp. 16–26. DOI: 10.1089/cmb.2018.0125.

[80] IGNACIO TINOCO, PHILIP N. BORER, BARBARA DENGLER, MARK D. LEVINE, OLKE C. UHLENBECK, DONALD M. CROTHERS, and JAY GRALLA. "Improved Estimation of Secondary Structure in Ribonucleic Acids." In: *Nature New Biology* 246.150 (1973), pp. 40–41. DOI: 10.1038/newbio246040a0.

[81] Melissa K. Takahashi and Julius B. Lucks. "A modular strategy for engineering orthogonal chimeric RNA transcription regulators." In: *Nucleic Acids Research* 41.15 (2013), pp. 7577–7588. DOI: 10.1093/nar/gkt452.

[82] Akito Taneda. "MODENA: a multi-objective RNA inverse folding." In: *Advances in Applied Bioinformatics Chemistry* 4 (2011), pp. 1–12.

[83] Akito Taneda. "MODENA: a multi-objective RNA inverse folding." In: *Adv Appl Bioinform Chem* 4 (2011), pp. 1–12. ISSN: 1178-6949. DOI: 10.2147/aabc.s14335.

[84] Ammar Tareen and Justin B Kinney. "Logomaker: beautiful sequence logos in Python." In: *Bioinformatics* 36.7 (2019). Ed. by Alfonso Valencia, pp. 2272–2274. DOI: 10.1093/bioinformatics/btz921.

[85] Youri Timsit and Sophie Bombard. "The 1.3 A resolution structure of the RNA tridecamer r(GCGUUUGAAACGC): metal ion binding correlates with base unstacking and groove contraction." In: *RNA (New York, N.Y.)* 13 (12 Dec. 2007), pp. 2098–2107. ISSN: 1469-9001. DOI: 10.1261/rna.730207. ppublish.

[86] Ignacio Tinoco and Carlos Bustamante. "How RNA folds." In: *J Mol Biol* 293.2 (1999), pp. 271–81. DOI: 10.1006/jmbi.1999.3001.

[87] Douglas H Turner and David H Mathews. "NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure." In: *Nucleic acids research* 38.suppl_1 (2010), pp. D280–D282.

[88] J. D. WATSON and F. H. C. CRICK. "Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid." In: *Nature* 171.4356 (1953), pp. 737–738. DOI: 10.1038/171737a0.

[89] Michael Waterman. "Secondary Structure of Single-Stranded Nucleic Acids." In: *Advances in Mathematics: Supplementary Studies* 1 (1978), pp. 167–212.

[90] 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. DOI: 10.1038/nprot.2006.249.

[91] Patrick C. Y. Woo, Yi Huang, Susanna K. P. Lau, and Kwok-Yung Yuen. "Coronavirus genomics and bioinformatics analysis." In: *Viruses* 2 (8 Aug. 2010), pp. 1804–1820. ISSN: 1999-4915. DOI: 10.3390/v2081803. ppublish.

[92] Alan R Woods. "Coloring rules for finite trees, and probabilities of monadic second order sentences." In: *Random Structures & Algorithms* 10.4 (1997), pp. 453–485.

[93] Sherry Y. Wu, Gabriel Lopez-Berestein, George A. Calin, and Anil K. Sood. "RNAi Therapies: Drugging the Undruggable." In: *Science Translational Medicine* 6.240 (2014), 240ps7. DOI: 10.1126/scitranslmed.3008362.

[94] Stefan Wuchty, Walter Fontana, Ivo L. Hofacker, and Peter Schuster. "Complete suboptimal folding of RNA and the stability of secondary structures." In: *Biopolymers* 49.2 (1999), pp. 145–165. DOI: 10.1002/(sici)1097-0282(199902)49:2<145::aid-bip4>3.0.co;2-g.

[95] Xiufeng Yang, Kazuki Yoshizoe, Akito Taneda, and Koji Tsuda. "RNA inverse folding using Monte Carlo tree search." In: *BMC bioinformatics* 18.1 (2017), p. 468. ISSN: 1471-2105. DOI: 10.1186/s12859-017-1882-7.

[96] Joseph N Zadeh, Brian R Wolfe, and Niles A Pierce. "Nucleic acid sequence design via efficient ensemble defect optimization." In: *Journal of Computational Chemistry* 32.3 (2011), pp. 439–52. DOI: 10.1002/jcc.21633.

[97] Shay Zakov, Yoav Goldberg, Michael Elhadad, and Michal Ziv-ukelson. "Rich Parameterization Improves RNA Structure Prediction." In: *Journal of Computational Biology* 18.11 (2011), pp. 1525–1542. DOI: 10.1089/cmb.2011.0184.

[98] Yu Zhou, Yann Ponty, Stéphane Vialette, Jérôme Waldispühl, Yi Zhang, and Alain Denise. "Flexible RNA design under structure and sequence constraints using formal languages." In: *ACM-BCB - ACM Conference on Bioinformatics, Computational Biology and Biomedical Informatics - 2013*. Bethesda, Washigton DC, United States, Sept. 2013. URL: https://hal.inria.fr/hal-00823279.

[99] M. Zuker and P. Stiegler. "Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information." In: *Nucleic Acids Research* 9 (1981), pp. 133–148.

[100] Michael Zuker and David Sankoff. "RNA secondary structures and their prediction." In: *Bulletin of Mathematical Biology* 46.4 (1984), pp. 591 –621. ISSN: 0092-8240. DOI: https://doi.org/10.1016/S0092-8240(84)80062-2. URL: http://www.sciencedirect.com/science/article/pii/S0092824084800622.

[101] mabseher. *htd*. https://github.com/mabseher/htd. July 2017.

ECOLE DOCTORALE

Titre : Décomposition Locale dans le Design Structural de l'ARN

Mots clés : Design négatif d'ARN, Complexité paramétrée, Motif d'ARN

Résumé : Le problème de design structural positif de l'ARN tente de trouver des séquences d'ARN réalisant une faible énergie libre de la structure secondaire cible. Par contre, dans le problème de design négatif, les séquences de solution doivent adopter la structure cible comme repliement préférentiellement à toute structure alternative. Le problème du repliement d'inverse, un problème typique de design négatif, exige que la cible soit la structure secondaire ayant l'énergie libre minimale (MFE) de la solution. D'autres métriques, telles que le défaut d'ensemble, sont également prises en compte pour l'évaluation de la séquence réalisée.
L'additivité du modèle d'énergie suggère l'existence de propriétés locales pour le problème de design de l'ARN. Il a été découvert dans plusieurs travaux que, en raison de la présence de certains motifs locaux, aucune séquence d'ARN ne peut se replier dans la structure cible tout en satisfaisant l'objectif de design négatif. L'approche d'échantillonnage de séquence est souvent utilisée dans le design positif. Les structures locales irréalisables, comme les paires de bases, se forment de manière répétée lors du repliement des séquences échantillonnées en considérant le design négatif. Dans cette thèse, nous étudions l'impact de cette nature locale sur l'aspect combinatoire et sur le développement de méthodes de design négatif.
Nous montrons que la proportion de structures secondaires réalisables diminue de façon exponentiellement avec la longueur de la structure cible du point de vue combinatoire. Étant donné une métrique de design négatif, nous proposons un schéma automatisé pour identifier tous les motifs non réalisables. L'énumération des structures secondaires évitant ces obstructions locales, suivie d'une analyse asymptotique, permet d'obtenir une borne supérieure du nombre de structures réalisables. En outre, nous définissons une borne inférieure pour le défaut d'ensemble structural dérivé des motifs locaux apparus. Nous montrons que cette borne inférieure suit une distribution limite Gaussienne avec une expression explicite, ce qui implique aussi la diminution exponentielle.
Nous présentons ensuite Infrared, un système générique d'échantillonnage combinatoire efficace. Nous formalisons le problème de design de l'ARN comme un problème de CSP avec des objectifs de design décrits comme un ensemble de contraintes et un ensemble de fonctions pondérées. Les évaluations des variables satisfaisant les contraintes sont générées à partir d'une distribution pondérée de Boltzmann en utilisant un algorithme de programmation dynamique suivi d'un backtrack stochastique. L'approche est en classe de FPT pour la largeur arborescente du graphe de dépendance induit par le problème. Nous montrons que ce cadre peut être facilement employé pour le design positif de l'ARN et les applications variées.
Enfin, en tant qu'application du système Infrared, nous proposons une approche originale d'échantillonnage itératif qui capture les principes de design négatif mis en œuvre dans RNAPOND. Un ensemble de paires de bases perturbatrices est identifié à chaque tour et on les empêche ensuite de s'apparier en introduisant des contraintes appropriées dans le cadre de l'échantillonnage. Malgré que le problème de décision associé est NP-difficile, un algorithme d'échantillonnage de séquence efficace est garanti par le système Infrared. Notre approche atteint un taux de réussite similaire ou supérieur aux états de l'art, tout en permettant la génération de séquences diverses et thermodynamiquement efficaces, c'est-à-dire des principes de design positif.
L'un des axes de recherche des travaux présentés dans cette thèse est l'extension à des structures plus complexes, telles que les structures secondaires contenant pseudonœuds. La flexibilité du système Infrared ouvre une porte au développement d'outils de design. Par exemple, le succès de RNAPOND suggère une approche potentielle pour la design structural négatif d'ARN.

Title : Local Decomposition in RNA Structural Design

Keywords : Negative RNA Design, Parametric Complexity, RNA motif

Abstract : RNA positive structural design problem attempts to find RNA sequences achieving low free energy of the target secondary structure. Differently, in the negative design, solution sequences should adopt the target structure as its folding preferentially to any alternative structure, according to the given metric and energy model. Inverse folding, a typical negative design, requires the target to be the solution sequence's MFE folding. Other metrics, like the ensemble defect, are also considered for design evaluation.
The additivity of the energy model suggests the existence of local properties for the RNA design problem. It was discovered in several works that, due to the presence of specific local motifs, some secondary structures are undesignable, *i.e.*, no RNA sequence can fold into the target structure while satisfying the negative design objective. The sequence sampling approach is often used in the positive design. Unwanted local structures, like base pairs, repeatedly form while folding sampled sequences toward the negative design. In this thesis, we study the impact of such local nature on the combinatorial aspect and on the development of negative design methods.
We show that the proportion of designable secondary structures decreases exponentially with the target structure length from the combinatorial aspect. Given a negative design metric, we propose an automated pipeline to identify all undesignable motifs. Enumerating secondary structures avoiding such local obstructions followed by asymptotic analysis yields an upper-bounds on the number of designable structures. In addition, we define a lower bound for the structural ensemble defect derived from occurred local motifs. We show that the lower bound follows a Normal limiting distribution with a closed-form expression, implying also an exponential decrease.
We then present Infrared, a generic framework for efficient combinatorial sampling. We formalize the RNA design problem as a CSP with design objectives described as a set of constraints and a set of weighted functions. Assignments satisfying constraints are generated from a Boltzmann weighted distribution using a dynamic programming algorithm followed by stochastic backtracking. The approach is FPT for the treewidth of the dependency graph induced from the problem. We show that the framework can be easily employed for RNA positive design and flexible applications.
Finally, as an application of Infrared, we propose an original iterative sampling approach that captures negative design principles implemented in `RNA POsitive and Negative Design` (RNAPOND). A set of DBPs is identified at each round and subsequently prevented from pairing by introducing proper constraints into the sampling framework. Despite the NP-hardness of the associated decision problem, an efficient sequence sampling algorithm is ensured by the Infrared framework. Our approach achieves a similar or better success rate than state-of-the-art negative design tools while allowing for the generation of diverse, thermodynamically efficient designs, *i.e.*, positive design principles.
One of the research directions of the works presented in this thesis is the extension to more complicated structures, such as pseudoknotted secondary structures. The flexibility of the Infrared framework opens a door for design tool development. For example, the success of RNAPOND suggests a potential approach for RNA negative structural design.

Institut Polytechnique de Paris
91120 Palaiseau, France